Optimization of basic algorithms in Commutative Algebra

PhD. Thesis

Massimo Caboara
Università degli Studi di Pisa
Dipartimento di Matematica
VIII Ciclo

29 May 1998

Contents

Introduction

This thesis has as its main theme the development of efficient algorithms for the computation of some basic operations in Commutative Algebra, e. g. ideal membership, projection, intersection, transporter, saturation, syzygies, resolution, radicals. Since the introduction of Gröbner bases by Buchberger [Bu], it has been possible to compute such operation effectively in practice, see Cox, Little, O'Shea [CLS] and Eisenbud [E] for an overview. Since the computations are often quite time-consuming, the development of efficient algorithms is a necessity. We are here mainly concerned with optimizations coming from Commutative Algebra, e. g. the use of Hilbert function, and not with optimizations coming from Computer Science (e. g. the data structures used) or Polynomial Arithmetic (e. g. the polynomial multiplication algorithm used). The optimizations presented here are meant for the "average" case. Hence, there is no proof of the effectiveness of the optimizations in the sense of Complexity Theory, but the algorithms are tested on meaningful examples. The algorithms discussed in this thesis don't require the input to be in "generic position", since our belief is that this constraint gives the best theoretical but worst practical complexities.

Gröbner basis computations are the main tool for the operations we are interested in, and they strongly depend on orderings. The orderings on polynomial rings are well known, see Robbiano [R1], but it has been necessary to develop a characterization of the orderings on a polynomial module that give the same ordering on every component, see Caboara and Silvestri [CS, CS1]. Since Gröbner basis computations are quite lengthy, we have developed two optimizations for the computation of a Gröbner basis:

> the use of the Hilbert function in multigraded cases, see Caboara, de Dominicis and Robbiano [CDR], a generalization of the work of Traverso [T], and
>
> dynamically changing the ordering during the computation of a Gröbner basis, see Caboara [Ca], [St].

Moreover we have developed a module version of the Buchberger algorithm as similar as possible to the usual ring algorithm. Such similarity (while retaining the efficiency of the original algorithm) is important for uniformity of implementation.

Syzygy computations are closely related to Gröbner basis computations, and a key step in several algorithms. We use the module Gröbner basis algorithm to compute syzygies by adding a tag variable. Moreover, with some modifications, this method allows us us to compute the minimal resolution of a graded module, see Caboara [Ca1]. Through our algorithm we are able to compute the (to my knowledge hitherto unknown) resolution of some well known hard examples, like the homogeneous "cyclic 7" ideal. An interesting discovery is that Schreyer's ordering for the syzygies in not always the best one.

The other key step for most basic algorithms is the intersection. We examine in detail the known algorithms, prove the substantial equivalence of the tag-variable and syzygy algorithms, and introduce the use of Hilbert Driven techniques, see Caboara and Traverso [CT]. Several examples hint that the Hilbert Driven algorithm is quite good. We then examine the transporter and saturation algorithms, and develop Hilbert Driven versions of them also.

In the last chapter, as an application of the algorithms described above, we give an algorithm for the equidimensional (if necessary, reduced) decomposition of an ideal. The computation of the radical is immediate from a reduced decomposition. We use as our main tool the projection along the coordinates (hence, Gröbner basis computation with respect to an elimination term-ordering). This method doesn't require the ideal to be in generic (dense) position. We use flatness to control pathological cases that arise from the lack of genericity, see Caboara, Conti and Traverso [CCT]. This algorithm seems to be at least the equal of the direct algorithms based on syzygy computations introduced by Eisenbud, Huneke and Vasconcelos [EHV].

In the Conclusions, we summarize the thesis's results and discuss the future development of the work started with this thesis.

We have aimed to make this thesis self-contained, the only prerequisites being some elementary Commutative and Computational Algebra, [AM, CLS, E, M, Mu, ZS]. Well known facts are usually recalled without proof. We give more details on more obscure topics, like the use of non-standard graduations, especially for Hilbert functions, and when we have to set up the notation for subsequent modifications. The original ideas expressed in this thesis are described in [Ca, Ca1, CDR, CS, CS1, CCT, CT].

At the end, in the section Problems and prospective, we discuss open problems and future developments.Below, we detail briefly the content and the results of each chapter.

Effective Rings and Modules (chapter 1)

This is an introductory chapter where some basic facts from Commutative Algebra are recalled; graduations and orderings are characterized, and an implementation for modules is discussed. We then examine in more detail the graduations given by integer matrices, especially from an effective point of view (e. g. how to find such matrices by linear algebra); see Robbiano [R2]. Throughout the thesis, "homogeneous" means "homogeneous with respect to one of these matrices". The characterization of ring orderings is then recalled, following Robbiano [R1], and a characterization of the module orderings which give the same ordering on each component is introduced, see Caboara and Silvestri [CS, CS1, Sv]. This is a special case of the more general characterization of module orderings, see Rust and Reid [RR]. Then we discuss an implementation of modules through an embedding in a suitable polynomial ring. Such an implementation is used throughout the thesis, to unify the treatment of ideals and modules; see Caboara and Traverso [CT] and Caboara [C]. Then there is a brief introduction to multigraded Hilbert function theory. Hilbert functions are a very important tool for the optimizations of our algorithms (see chapter for 2 to 5). Our interest lies mainly in the computation of Hilbert functions, and in some formulas that are needed for the applications in the subsequent chapters.

Gröbner Bases (chapter 2)

In this chapter we recall briefly some basic facts of Gröbner basis theory, and give some variants of the Buchberger Algorithm for Gröbner basis computation. For an overview of Gröbner basis theory, see Buchberger [Bu], Cox, Little and O'Shea [CLS] and Eisenbud [E]. We examine the homogeneous version of the Buchberger algorithm, mainly to set up notation, and discuss the use of Hilbert functions to optimize the Buchberger algorithm, see Traverso [T]. We then generalize the Traverso Hilbert-Driven algorithm to the multigraded case, see Caboara, de Dominicis and Robbiano [CDR]. Some examples show the utility of this "Bi-Hilbert-Driven algorithm" when it is possible to apply it. A version of the Buchberger algorithm that changes dynamically the ordering during the computation is then discussed, see Caboara [Ca] and Sturmfels [St]. The feasibility tests for each ordering are performed using the simplex algorithm, while the heuristics that select the "optimal" ordering are based on Hilbert function considerations. We then briefly discuss a module version of the Buchberger algorithm that uses the embedding introduced in chapter 1. Since this module Buchberger algorithm differs from the usual Buchberger algorithm only by the definition of some internal procedures, (mainly the division test for terms), from here on there will be no special treatment for modules/submodules. Finally, a version of the Buchberger algorithm that factorizes the polynomials of the Gröbner basis is briefly discussed. A factorizing Buchberger algorithm is useful for the ideal decompositions algorithms of chapter 5.

Syzygies (chapter 3)

In this chapter we examine briefly the computation of a minimal basis for an homogeneous module, mainly to set up the notation for the minimal resolution algorithm. The two known algorithms for syzygy computations are then compared, and it is found that they are equal, except for the timing of some sub-operation, see Caboara and Traverso [CT]. Then, the tag-variable syzygy algorithm and the algorithm for the computation of a minimal basis of a module are combined to give an algorithm for the computation of a minimal resolution of a graded module, see Caboara [Ca1]. This algorithm is quite efficient, as shown by some examples, and on the average on a par with the more recent algorithms, see Capani, de Dominicis, Niesi, Robbiano [CDNR] and La Scala, Stillman [LS]. Moreover, using the characterization of module orderings given in chapter 1, it is quite easy to explore different strategies With the resolution algorithm it has been possible to compute for the first time some quite complex resolutions, like that of the "cyclic 7" ideal, see section 3.7. A main feature of the resolution algorithm is its ease of implementation, since it is only a slight modification of the usual Buchberger algorithm. We prove termination and correctness of the algorithm.

Operations (chapter 4)

In this chapter we examine in detail the implementation of three basic operation of computational ideal/module theory : intersection, transporter and saturation. Several algorithms for the ideal and module cases are analyzed and compared. Since most algorithms use intersection as a key step, the algorithms for intersection are examined with special care. In particular, the tag variable algorithm and the syzygy algorithm for intersections are proved equivalent. We develop an Hilbert-Driven version of the intersection algorithm that seems to be quite efficient, see section 4.4.5. The transporter and saturation algorithms are then studied. The same techniques used in the intersection case gives us a Hilbert-Driven version of the transporter and saturation algorithm.

Ideal Decompositions (chapter 5)

In this chapter we develop an algorithm that gives an equidimensional decomposition of an ideal. The equidimensional decomposition is a first step for the reduced decomposition, (and also, radical computation), and primary decomposition. We give also the algorithm for the computation of a reduced decomposition from an equidimensional decomposition. The main tool is the projection of the variety to a hypersurface along coordinates (from the algebraic point of view, computation of Gröbner bases with respect to an elimination ordering). We use flatness to control the pathological cases that arise using non-generic projections. When such a case is detected, the ideal is decomposed in two or more simpler ideals through GCD s and saturations. Full use is made of the optimized operation introduced in chapter 4, and of the Hilbert Driven techniques of chapter 2. Since the ideals are kept sparse, the algorithm is quite efficient, see the examples in section 5.5. This algorithm seems to be a valid alternative to the direct algorithms introduced by Eisenbud, Huneke and Vasconcelos [EHV], that use syzygy and resolution computations as their main tool. This chapter is intended for showing the use of the algorithms developed in the previous chapters as basic steps for a more complex, non-basic algorithm. Hence, the analysis is not as detailed as in the other chapters. A full treatment of the decomposition algorithms requires a thesis in itself.

When not explicitly said, the computations have been performed with a PentiumPro, 200Mz, 96MB RAM, working under Linux 2.0.30. The CoCoA versions are `cocoa3011b` or `cocoa3.3`, [COCOA]. The performance of the two versions are quite similar. We have employed both because `cocoa3.3` has a more complete treatment of the Hilbert-Driven algorithms, while `cocoa3011b` is more stable.

Chapter 1

Computational rings and modules

In this chapter we briefly detail the kinds of rings and modules that are of computational interest from our point of view. The main points are :

a description of graded rings and modules from a computational point of view, section 1.1;

a description and a partial classification of the orderings for rings and modules, section 1.2 and 1.3;

the Hilbert Function and the algorithms to compute it, section 1.4.

1.1 Graded Rings and Modules

The standard work is [ZS]; see [R2] for a description of non-standard graduations.

1.1.1 Graded rings

Definition 1.1 *Let A be a ring, G be a semigroup and $\mathcal{A} := \{A_d\}_{d\in G}$ be a family of subgroups of A such that*

- $A = \bigoplus_{d\in G} A_d$.
- $\forall a \in A_d, b \in A_{d'}$ *we have* $ab \in A_{d+d'}$.

The family $\mathcal{A}$ is a graduation *of A over G and A is* graded *over G. The subgroup A_d, $d \in G$ is the* homogeneous component of G of degree d *and the elements $a \in A_d$ are the* homogeneous elements of degree d of A.

Remark 1.2 *We note that if A is a graded ring with graduation $\mathcal{A} := \{A_t\}_{t\in G}$, and 0 is the identity of G, then 0 is an homogeneous element of A_t for each $t \in G$, A_0 is a subring of A and A_t is an A_0 module for each $t \in G$.*

1.1.2 Graded modules

Definition 1.3 *Let A be a graded ring with graduation $\mathcal{A} := \{A_t\}_{t\in G}$ and M be an A module. Let $\mathcal{M} := \{M_d\}_{d\in G}$ be a family of subgroups of M such that*

- $M = \bigoplus_{d\in G} M_d$.
- $am \in M_{t+d} \ \ \forall \ \ a \in A_t, \ m \in M_d$.

is graded *over* G. *The submodule* M_d, $d \in G$ *is the* homogeneous component of degree d of M *and the elements* M_d *are* homogeneous elements of degree d of M..

Remark 1.4 *For every element* $m \in M$*, there is one and only one way to write* m *as sum of homogeneous elements of* M*, except for the ordering of the elements of the sum.*

Proposition-Definition 1.5 *Let* A *be a ring and* $(M, \mathcal{M})$ *be an* R *module graded over the semigroup* G*. Then a submodule* $N \subset M$ *is* homogeneous *if, equivalently,*

- $N = \bigoplus_{d \in G}(N \cap M_d)$*, or*
- $\forall m = \sum_{d \in G} m_d, \quad m \in N \Longleftrightarrow m_d \in N \; \forall \; d \in G$*, or*
- N *is generated by homogeneous elements.*

Proof. This is standard, see p.e. [M]. □

Definition 1.6 *Let* $(A, \mathcal{A})$ *be a graded ring and* I *be an ideal of* A*. Then* I *is homogeneous as ideal if it is homogeneous as submodule of* $(A, \mathcal{A})$*.*

Definition 1.7 *Let* $\mathcal{A}_i := \{$ *polynomials of degree* $\leq i\}$*. Then* $\{\mathcal{A}_i\}_{i \in \mathbb{N}}$ *is the* standard graduation *for* $k[x_1, \ldots, x_n]$ *over* $\mathbb{N}$.

To ease the notation, when there is not risk of ambiguity, the mention of the graduation and of the subgroup would be omitted.

1.1.3 Graded morphisms

Definition 1.8 *Let* $(M, \mathcal{M})$*,* $(N, \mathcal{N})$ *be graded* A *modules over* G *and* $f : M \to N$ *be an* A *module morphism. The morphism* f *is* graded (of degree $t \in G$) *if* $f(M_d) \subseteq N_{d+t} \; \forall d \in G$*.*

Definition 1.9 *An exact sequence of graded modules is* graded of degree t *if all its morphism are graded of degree* t*.*

Example 1.10 *Let* $k[x]$ *be graded with the standard graduation. Then the morphism* $id_{k[x]} : k[x] \to k[x]$*, is graded of degree* 0*, and the morphism* $k[x]@ > \cdot x >> k[x]$*, is graded of degree* 1*.*

1.1.4 Shifts

Definition 1.11 *Let* M *be a graded* A *module with graduation* $\mathcal{M} := \{M_d\}_{d \in G}$ *and* $t \in G$*. The family* $\{M'_d\}_{d \in G} := M_{d+t}$ *is a graduation of* M *and is the graduation* $\mathcal{M}$ shifted *by* t*. The standard graduation of* M *shifted by* t *is denoted as* $M(t)$*.*

Example 1.12 *Let* $M := k[x, y]$*. Then* $M(-1)_0 := (0)$*,* $M(-1)_1 := (0, 1)$*,* $M(-1)_2 := (0, x, y)$*,* $M(-1)_3 := (0, x^2, xy, y^2)$*, ...*

Example 1.13 *Let* $k[x]$ *be graded with the standard graduation. Then the morphism*

$$\begin{array}{ccc} k[x](-1) & @ > \cdot x >> & k[x] \\ f & \rightsquigarrow & xf \end{array}$$

is graded of degree 0*.*

1.1.5 Some results for graded rings/modules

Proposition 1.14 *Let M be a graded A module and N be an homogeneous submodule of M. Then the A module M/N is graded with graduation $(M/N)_d := M_d/N_d$*

Proof. Easy check. □

Proposition 1.15 *Let I and J be homogeneous ideals of a graded ring A. Then the ideal*

a) $I + J$ is homogeneous;

b) $I \cup J$ is homogeneous;

c) $I : J$ is homogeneous;

c) $I : J^\infty$ is homogeneous.

d) $\sqrt{I}$ is homogeneous.

e) The prime ideals associated to I are homogeneous.

f) There exist $Q_1, \dots, Q_v$ homogeneous primary ideals such that $I := Q_1 \cap \dots \cap Q_v$.

Proof. We prove only *d)*, the other cases are similar.
Let τ be a total ordering over G compatible with the semigroup structure. By the remark 1.4, since $a \in R$, $a = a_{d_1} + \dots + a_{d_t}$, where $d_1 <_\tau \dots <_\tau d_t$. By the proposition 1.5, it suffices to prove that if $a \in \sqrt{I}$ then $a_{d_i} \in \sqrt{I}$ for each $i : 1 \dots t$. But $a^n = a^n_{d_1} + \dots$ elements of degree $>_\tau nd_1$; since I is homogeneous $(a^n)_{nd_1} = a^n_{d_1}$, thus $a^n_{d_1} \in I$ and hence $a_{d_1} \in \sqrt{I}$. Thus $a - a_{d_1} \in \sqrt{I}$, and the thesis is proved by iteration. □

Proposition 1.16 *Let A be a graded ring and M a graded A module. Let N and P be homogeneous submodules of M. Then*

i) the A module $N \underset{M}{:} P := \{m \in M \mid mP \subseteq N\}$ is homogeneous.

ii) The ideal $N \underset{A}{:} P := \{a \in A \mid aP \subseteq N\} \subset A$ is homogeneous.

Proof. The proofs of *a)* and *b)* are similar to the proof of *a)* of proposition 1.15. □

1.1.6 Effective graduations

Definition 1.17 *Let G be a semigroup. A graduation $\mathcal{A}$ over $R := k[x_1, \dots, x_n]$ is geometric if*

a) $k \subset R_0$

b) the variables $x_1, \dots, x_n$ are homogeneous.

Corollary 1.18 *Let $\mathcal{A}$ be an geometric graduation over $R := k[x_1, \dots, x_n]$. Then*

a) every term of R is homogeneous.

b) R_t is an R_0 vectorial space for each $t \in G$.

Definition 1.19 *Let $R := k[x_1, \dots, x_n]$ be a ring with geometric graduation $\mathcal{A}$ over $\mathbb{Z}^r$. Then it is possible to build a matrix $\mathbf{w} \in M_{r \times n}(\mathbb{Z})$, where $(w_{1,i}, \dots, w_{r,i}) := deg(x_i)$. The i^{th} column of the matrix $\mathbf{w}$ is the* (multi)degree *or* (multi)weight *of the variable x_i, sometimes denoted as $\partial_{\mathbf{w}}(x_i)$. The ring R is* graded by $\mathbf{w}$ over $\mathbb{Z}^r$*, and the vectorial space of the homogeneous polynomials of R of degree $\mathbf{d} := (d_1, \dots, d_r)$ is $(R, \mathbf{w})_{\mathbf{d}}$.*

Proposition 1.20 *Let $R := (k[x_1, \dots, x_n], \mathbf{w})$ be a graded ring. If there exists a linear combination over $\mathbb{Z}$ of the rows of $\mathbf{w}$ such that its elements are positive, then $\#(R, \mathbf{w})_{\boldsymbol{d}} < \infty$ for each $\mathbf{d} \in \mathbb{Z}^r$.*

Proposition 1.21 *Let $R := (k[x_1, \dots, x_n], \mathbf{w})$ be a graded ring. Then, if there exists a submatrix $\mathbf{w}'$ of $\mathbf{w}$ such that*

1. *$\mathbf{w}' \in M_{r',n}(\mathbb{Z}), r' \leq r$;*
2. *the entries of $\mathbf{w}'$ are non negative;*
3. *every column of $\mathbf{w}'$ has a positive entry.*

We have that $\#(R, \mathbf{w})_{\boldsymbol{d}} < \infty$ for each $\mathbf{d} \in \mathbb{Z}^r$.

Definition 1.22 *A matrix $\mathbf{w}$ fulfilling the conditions of proposition 1.20 or 1.21 is a* system of weights *for R and gives a geometric graduation for R over $\mathbb{Z}^r$. In the following, every graduation will be geometric over $\mathbb{Z}^r$, for some $r \geq 1$,and will be given through a suitable system of weights $\mathbf{w}$. The number of linearly independent rows of $\mathbf{w}$ is the* type *of $\mathbf{w}$.*

Definition 1.23 *The system of weights $\mathbf{u}$ and $\mathbf{w}$ are* equivalent *if their associated graduations are equivalent. A system of weights $\mathbf{w}$ is the* refinement *of a system of weights $\mathbf{u}$ of type m if $w_1 = u_1, \dots, w_m = u_m$.*

Remark 1.24 *The order of the rows of a system of weights $\mathbf{w}$ for R is arbitrary in the sense that changing the order we obtain a system of weights which induces the same graduation. To a system of weights are associated two natural ordering on the set of the homogeneous components*

Definition 1.25 *Let $(R, \mathbf{w})$ be a graded ring. Then the two orderings on the set of the homogeneous components of $R, \mathbf{w}$, σ_1, σ_2 defined by*

- $R_{\mathbf{d}} >_{\sigma_1} R_{\mathbf{d}'} \iff (d_1, \dots, d_r) >_{lex} (d'_1, \dots, d'_r)$
- $R_{\mathbf{d}} >_{\sigma_2} R_{\mathbf{d}'} \iff d_i \geq d'_i$ *for each* $i : 1 \dots r$

are respectively the $\mathbf{w}$ degree *ordering and the $\mathbf{w}$* uniform degree *ordering.. The $\mathbf{w}$ uniform degree ordering is usually a partial ordering, while the $\mathbf{w}$ degree ordering is total. Since the set of the homogeneous component of $R, \mathbf{w}$ in indexed by $\mathbb{Z}^r$, any term-ordering on $\mathbb{Z}^r$ is a term-ordering on it.*

Some interesting graduations :

Example 1.26 *Let $\mathbf{w} := [\underbrace{1, \dots, 1}_{n}] \in \mathbb{Q}$. Then the graduation given by $\mathbf{w}$ on $k[x_1, \dots, x_n]$ is the standard graduation.*

Example 1.27 *Let $\mathbf{w} \in M_{n \times n}(\mathbb{Z})$ be a system of weights for $k[x_1, \dots, x_n]$. Then the graduation given by $\mathbf{w}$ over $k[x_1, \dots, x_n]$ is the* fine graduation . *The homogeneous components of $k[x_1, \dots, x_n], \mathbf{w}$ are the sets $\{t\}$, $t \in T^n$.*

Corollary 1.28 *From proposition 1.15 and example 1.27 it follows that the sum, intersection, radical and saturation of monomial ideals are monomial ideals. Similarly, the associated primes of a monomial ideal are monomial ideals.*

It is quite easy to determine the existence, and to build, if it exists, a graduation of $k[x_1, \dots, x_n]$ such that a given polynomial would be homogeneous, if necessary of a given degree.

Proposition 1.29 *Let $R := k[x_1, \ldots, x_n]$ and $f := \sum_i a_i t_i \in k[x_1, \ldots, x_n]$ with $a_i \in k$. Let $X_{i,j}$ be indeterminates and $\mathbf{w} := (X_{i,j}),\ i : 1 \ldots n,\ j : 1 \ldots r$ Then, a solution of the system*

$$\begin{cases} \mathbf{w} \cdot \mathrm{Log}\,(t_1) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_2) \\ \mathbf{w} \cdot \mathrm{Log}\,(t_1) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_3) \\ \vdots & \vdots & \vdots \\ \mathbf{w} \cdot \mathrm{Log}\,(t_1) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_k) \end{cases} \tag{1.1}$$

with respect to the indeterminates $X_{i,j}$ that fulfill the conditions of proposition 1.20 give a system of weights for R such that f is homogeneous with respect to it.

Proof. Immediate by the definition of system of weights. □

Example 1.30 *Let $f := x^2 - x$ be a polynomial of $k[x]$. It is easy to see that no system of weights on $k[x]$ gives a graduation with respect which f is homogeneous.*

Since not all polynomials may be made homogeneous by changing only the system of weights, it makes sense to generalize the above procedure. By adding a sufficient number of variables, any set of polynomials may be made homogeneous, often even with restrictions on the system of weights matrix.

Proposition 1.31 *Let $R := k[x_1, \ldots, x_n]$, $f := \sum_{i:=1}^k a_i t_i \in k[x_1, \ldots, x_n]$ and $\mathbf{h} := \{h_1, \ldots, h_r\}$ new variables. Let $X_{i,j}, Y_{l,s}$ be indeterminates with $i : 1 \ldots n+r,\ j, s : 1 \ldots r,\ l : 1 \ldots k$ and $\mathbf{w} := (X_{i,j})$ Then, the solutions of the system*

$$\begin{cases} \mathbf{w} \cdot \mathrm{Log}\,(t_1 \mathbf{h}^{Y_1}) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_2 \mathbf{h}^{Y_2}) \\ \mathbf{w} \cdot \mathrm{Log}\,(t_1 \mathbf{h}^{Y_1}) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_3 \mathbf{h}^{Y_3}) \\ \vdots & \vdots & \vdots \\ \mathbf{w} \cdot \mathrm{Log}\,(t_1 \mathbf{h}^{Y_1}) & = & \mathbf{w} \cdot \mathrm{Log}\,(t_k \mathbf{h}^{Y_k}) \end{cases} \tag{1.2}$$

with respect to the indeterminates $X_{i,j}$ that fulfill the conditions of proposition 1.20 give a system of weights for R such that the polynomial $\sum_{i:=1}^k a_i t_i \mathbf{h}^{Y_l}$ is homogeneous.

Proof. Immediate by the definition of system of weights. □

Corollary 1.32 *Let $\mathbf{u}$ be a system of weights on R. Then for each polynomial $f \in R$ and for each $r : 1 \ldots n$ there exists a system of weights $\mathbf{w}$ of type r on $R[\mathbf{h}]$ such that f is $\mathbf{w}$ homogeneous and $\mathbf{w}$ restricted to R is equal to $\mathbf{u}$.*

Proof. If f is homogeneous for some system of weights $\mathbf{u}$, we discover such $\mathbf{u}$ using proposition 1.29. If not, we follow the following procedure: we take

$$X_{i,j} = \begin{cases} \delta_{i,j} & \text{for } i : 1 \ldots n \text{ and } j : n+1 \ldots n+r \\ u_{i,j} & \text{for } i : 1 \ldots n \text{ and } j : 1 \ldots n \end{cases}$$

and we order the indeterminates by an ordering $>$ such that $Y_1 > \ldots > Y_r > X$. Then, it is immediate to check that the system (1.2) is an upper triangular homogeneous system of $(n + r - 1) \times k$ equations in $n \times r + r \times k$ indeterminates whose associated matrix is

$$\left[\begin{array}{c|c|c} & -1 & \\ \mathbf{I_{r-1}} & \vdots & \mathbf{A} \\ & -1 & \end{array}\right] \tag{1.3}$$

where $\mathbf{A} \in \mathrm{Mat}_{n \times k}(k)$. Such system has obviously positive integer solutions. Let $\mathbf{w}$ be the matrix of the X solutions of (1.3). The polynomial $\sum_{i:=1}^k a_i t_i \mathbf{h}^{Y_l}$ is then homogeneous with respect to the system of weights $\mathbf{w}$. □

Remark 1.33 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{u}$ be a system of weights for R of type r and $f \in R$. Then corollary 1.32 gives the system of weights*

$$\mathbf{w} := \left[\begin{array}{c|c} \mathbf{u} & I \end{array} \right]$$

for $R[\mathbf{h}]$ and a procedure to build a polynomial f' such that

f' is $\mathbf{w}$ homogeneous and

$f'(x_1, \dots, x_n, 1, \dots, 1) = f(x_1, \dots, x_n)$. Indeed

$$f = \sum_{i=1}^{k} a_i t_i \rightsquigarrow \sum_{i=1}^{k} a_i t_i h_1^{\partial_{u_1}(t)} \cdots h_r^{\partial_{u_r}(t)} =: f'$$

Example 1.34 *Let $R := k[x, y, z]$ and $\mathbf{u} := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 0 & 0 & 2 \end{bmatrix}$. Then*

$$\mathbf{w} := \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 4 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and
$f := z^4 + xy + z \rightsquigarrow z^4 h_3^2 + xy h_1^2 h_2^{13} + z h_1^3 h_2^{12} h_3^2 =: f^{h,\mathbf{u}}$.

Definition 1.35 *Let $R := k[x_1, \dots, x_n]$ and $f \in R$. Let $h_1, \dots, h_r$ be new variables. Let $f' \in R[\mathbf{h}]$ be such that*

1. *there exists a system of weights $\mathbf{w}$ on $R[\mathbf{h}]$ such that f' is $\mathbf{w}$ homogeneous;*
2. *$f'(x_1, \dots, x_n, 1, \dots, 1) = f(x_1, \dots, x_n)$.*

Then the couple $(f', \mathbf{w})$ is an homogeneization *of f. If $\mathbf{u}$ is a system of weights on R and the restriction of $\mathbf{w}$ to R is equivalent to $\mathbf{u}$, then f' is* an homogeneization *of f with respect to $\mathbf{u}$. If f' is the polynomial produced by the operation described in remark 1.33, then it is* the homogeneization *of f with respect to $\mathbf{u}$.*

Corollary 1.32 gives a solution, not necessarily a minimal solution with respect to the number of new variables introduced. A way to add fewer variables is to try to solve system (1.2) introducing only one new variable and, if there are no acceptable solutions, introduce two variables and so on.

Homogeneization of set of polynomials/module elements follows straightforwardly from the homogeneization of a single polynomial. Homogeneization of ideals/modules is more complex, see section 4.3.

Proposition 1.36 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights and M be a free, finitely generated R module of rank k. Then a matrix $\mathbf{w}' \in \mathrm{Mat}_{(n+k) \times r})$ gives a graduation on M. All the proposition above holds for such modules, and the notation used for rings applies to modules. The graduations that give the same degree to the same term independently from the component in which the term is are* Riquier graduations. *For each Riquier graduation there is a matrix $\mathbf{w}' \in \mathrm{Mat}_{(n+k) \times r})$ that gives an equivalent graduation.*

Proof. That a matrix $\mathbf{w}' \in \mathrm{Mat}_{(n+k) \times r})$ gives a Riquier graduation is obvious. The converse is easy after the classification of module orderings, see section 1.3. Refer to the example below for a hint. □

Example 1.37 *Let $R := k[x, y, z, t]$, $\mathbf{w} := \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 2 & 3 \end{bmatrix}$ and $M := R^2$. Then if we give degree $(3, 0)$ to $\mathbf{e}_1$ and degree $(0, 1)$ to $\mathbf{e}_2$ the element $x^3 y^2 \mathbf{e}_1 + y^7 z \mathbf{e}_2$ is homogeneous with respect to the matrix*

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 3 & 0 \\ 0 & 0 & 2 & 3 & 0 & 1 \end{bmatrix}$$

1.2 Ring orderings

The term-orderings on a polynomial ring are closely related with the computation of Gröbner bases and the efficiency of Buchberger Algorithm; they are classified by Robbiano [R1].

The module orderings are substantially more complex. Module orderings which give the same ordering on every component are classified here, see also [S, CS]. For more general orderings, we refer to [CSi] and [RR].

Definition 1.38 *Let $R := k[x_1, \dots, x_n]$. A total ordering τ on T^n such that $t > 1 \ \forall\, t \in T^n$ is a* term-ordering. *The set of all term-orderings on R is* $\mathrm{TO}(T^n)$.

Remark 1.39 *A term-ordering τ on a ring $R := k[x_1, \dots, x_n]$ has a crucial finiteness property: There are no infinite strictly τ decreasing sequences of terms of R*

Definition 1.40 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(T^n)$ and $f \in R$. Then the term $Lt_\tau(f) := Max_\tau(Supp(f))$ is the* leading term *of f with respect to τ. Similarly, $Lm_\tau(f)$ is the* leading monomial *of f. The coefficient in f of $Lt_\tau(f)$ is the* leading coefficient *di f with respect to τ, or $Lc_\tau(f)$. To ease the notation, when no ambiguity arises, there would be no explicit reference to the term-ordering.*

Definition 1.41 *Let A and B be rings and $\tau \in T(A)$, $\sigma \in T(B)$. Then if the map $f : A@>>> B$ is such that $a >_\tau b \Longrightarrow f(a) >_\sigma f(b)$, then f is* ordered.

Remark 1.42

i) An ordering (term-ordering) on T^n may be easily extended to an ordering (term-ordering)on R.

ii) An ordering (term-ordering)on $\mathbb{N}^n$ may be easily extended to an ordering (term-ordering)on $\mathbb{Z}^n$.

iii) An ordering on $\mathbb{Z}^n$ may be easily extended to an ordering (term-ordering)on $\mathbb{Q}^n$.

Hence, to give an ordering (term-ordering) on $k[x_1, \dots, x_n]$ it suffices to give an ordering (term-ordering) on $T^n \simeq \mathbb{N}^n \subset \mathbb{Z}^n$, thus on $\mathbb{Z}^n$.

Theorem 1.43 *Let τ be a total ordering on $\mathbb{Q}^n$. Then there exists a list $\mathbf{u} := (u_1, \dots, u_s)$ of rationally independent vectors in $\mathbb{R}^n$ such that*

a) The map

$$\begin{array}{cccc} \alpha : & (\mathbb{Q}^n, \tau) \quad @> & >> & (\mathbb{R}^s, lex) \\ & v & \rightsquigarrow & (v \cdot u_1, \dots, v \cdot u_s) \end{array}$$

is linear, ordered and injective.

b) If $d(u_i)$ is the dimension of the rational vector space spanned by the coordinates of u_i, then

$$\sum_{i=1}^{s} d(u_i) = n.$$

If τ is a term-ordering, then

c) the first non zero entry for every component is positive.

Vice versa, every list $[u_1, \dots, u_s]$ of vectors which satisfy the conditions a) *and* b) *defines a total ordering on $\mathbb{Q}^n$. If $\mathbf{u}$ fulfill condition* c) *too, then it defines a term-ordering.*

Definition 1.44 *If $u_i \in \mathbb{Q}^n$ for each $i : 1 \dots n$, it is useful to view the vector list $\mathbf{u}$ as a matrix, whose rows are $u_1, \dots, u_s$. In this case, $Det(\mathbf{u})$ has to be non zero to fulfill* b). *Such orderings are* rational orderings. *Matrices associated to rational term-orderings have the first non null entry of each column positive. If τ is an ordering, then the associated matrix is often denoted as Γ_τ.*

Remark 1.45 *Partial orderings are associated to vector lists such that* $\sum_{i=1}^{s} d(u_i) < n$. *Partial orderings such that* $t > 1 \ \forall\, t \in T^n$ *(*partial term-orderings*) are associated to vector lists such that* $\sum_{i=1}^{s} d(u_i) < n$, *but that fulfill condition* c).

Definition 1.46 *With the notation of theorem 1.43, the (vector list) matrix* $\mathbf{u}$ *is* associated to *or* gives *the term-ordering* τ.

Remark 1.47 *Different matrices may give the same term-ordering. In fact, if to the term-ordering* τ *is associated the matrix* $\mathbf{u} := (u_1, \dots, u_n)$, *any matrix* $\mathbf{w}$ *whose rows* w_j *are a linear combination of the rows* u_i, $i : 1 \dots j$ *is associated to* τ. *We refer to Schwartz [Sw] for an analysis of the matrices associated to term-orderings.*

Proposition-Definition 1.48 *If* $\mathbf{u}$ *is a matrix associated to the ordering* τ, *then any matrix associated to* τ *has as first row a multiple of* u_1 *The positive half-line spanned by* u_1 *in* $(\mathbb{R}^n)^+$ *is the* half line associated to τ. *The* first vector *of* τ *is the positive versor of the half line associated to* τ.

Example 1.49 *The vector list* $[(1, \sqrt{37})]$ *defines a term-ordering* τ *on* $k[x, y]$ *such that* $y > x^3$. *The checking is easy,* $y >_\tau x^3 \Leftrightarrow (-3, 1)(1, \sqrt{37}) > 0 \Leftrightarrow -3 + \sqrt{37} > 0$. *The positive half-line associated to* τ *has equation* $\sqrt{37}y - x = 0$, *and the first vector of* τ *is* $(1, \sqrt{37})/\sqrt{38}$.

1.2.1 Some interesting orderings

Definition 1.50 *Let* $R := k[x_1, \dots, x_n]$. *The term-ordering* τ *associated to a vector list which is a permutation of the vector list* $[\mathbf{e}_1, \dots, \mathbf{e}_n]$ *is a* lexicographic ordering. *If* τ *is termed lex ordering for* R *without further specification, it is supposed that* $x_1 >_\tau x_2 >_\tau \dots >_\tau x_n$.

Example 1.51 *Let* $R := k[x, y, z]$. *Then the term-ordering associated to the matrix*

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

is the lex *term-ordering.*

Definition 1.52 *Let* $R := k[x_1, \dots, x_n]$, $V := \{x_i\}_{i \in W}$, $\tau \in \mathrm{TO}(k[V])$ *and* $\sigma \in \mathrm{TO}(k[\mathbf{x} - W])$. *If* $\Gamma_\tau, \Gamma_\sigma$ *are matrices associated to* τ *and* σ, *then the term-ordering on* R *associated to the matrix*

$$\begin{bmatrix} \Gamma_\tau & \mathbf{0} \\ \mathbf{0} & \Gamma_\sigma \end{bmatrix}$$

is the block ordering, *or* elimination ordering *(for* V*)* $\tau \times \sigma$. *If* σ *is an ordering, and* $V \subset \{x_1, \dots, x_n\}$ *then* σ_V *is the restriction of* σ *to* $k[V]$.

Remark 1.53 *The lex term-ordering introduced in definition 1.50 is an elimination ordering for* $\{x_1\}, \dots, \{x_1, \dots, x_j\}, , \{x_1, \dots, x_{n-1}\}$. *Elimination orderings have strong algebro-geometric properties, see chapter 4 and 5.*

Remark 1.54 *A system of weights is defined by a matrix, and the order of the rows doesn't matter (a permutation of the rows gives an equivalent system of weights). Hence a couple (*$\mathbf{w}$, *an ordering on the rows of* $\mathbf{w}$*) gives a partial term-ordering on* R.

Definition 1.55 *Let* $R := k[x_1, \dots, x_n]$ *be a ring graded by the system of weights* $\mathbf{w}$ *of type* r. *A term-ordering on* R *such that the first* r *rows of its associated matrix are linearly dependent from* $\mathbf{w}$ *is a term-ordering* compatible with to $\mathbf{w}$.

Example 1.56 *Let $S := k[u, v, w, l]$ be graded by the system of weights*

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

Then the term-ordering σ associated to the matrix

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

is compatible with $\mathbf{w}$.

Example 1.57 *Let R, τ and S, σ be the ordered rings described in the examples 1.51 and 1.56. Then the term-ordering $\tau \times \sigma$ on $k[x, y, z, u, v, w, l]$ is associated to the matrix*

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

Definition 1.58 *Let $R := k[x_1, \ldots, x_n]$, τ be a term-ordering on R associated to the vector list whose first element is $(1, \cdots, 1)$ and the others a permutation of $n-1$ of the vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$. The term-ordering τ is a* deglex *ordering on R. If τ is called a deglex ordering for R without further specification, it is supposed that $x_1 >_\tau x_2 >_\tau \ldots >_\tau x_n$.*

Definition 1.59 *Let $R := k[x_1, \ldots, x_n]$ and τ a term-ordering on R associated to the vector list whose first element is $(1, \cdots, 1)$ and the others a permutation of $n-1$ of the vectors $-\mathbf{e}_1, \ldots, -\mathbf{e}_n$. The term-ordering τ is a* degrevlex *ordering on R. If τ is called a deglex ordering for R without further specification, it is supposed that $x_1 <_\tau x_2 <_\tau \ldots <_\tau x_n$.*

Gröbner basis computations with respect to the degrevlex ordering give usually the best timings, see [BS2].

The degrevlex ordering is usually quite good for the computation of Gröbner bases (see chapter 2). Hence, when there is some freedom in the the choosing of an ordering, it is usually a good idea to choose it as similar as possible to the degrevlex ordering. So, when we speak of *the* completion of an ordering we intend the ordering that has the maximum possible number of rows equal to $-\mathbf{e}_i$, with the indexes i has big as possible. So, when we speak of *the* block-ordering we intend the block ordering whose blocks defines degrevlex orderings, and so on.

1.2.2 Semi flags

We introduce some notions that will be useful in the following for the classification of module term-orderings.

Definition 1.60 *Let τ be a ordering on T^n. The set of positive terms of τ is the set*

$$P_\tau := \{\boldsymbol{a} \in \mathbb{N}^n \mid x^{\boldsymbol{a}} >_\tau 1\}.$$

Note that giving the set of positive terms of an ordering is a way to define it.

Remark 1.61 *If* $\mathbf{u} := [u_1, \ldots, u_s]$ *is a list of vectors defining the ordering* τ*, then the set of positive terms of* τ *is the subset of the points* $p \in \mathbb{Z}^n$ *satisfying at least one of the following systems of equations/inequalities*

$$\left\{ \begin{array}{rcl} u_{11}x_1 + \cdots + u_{1n}x_n & > & 0 \end{array} \right.$$

$$\left\{ \begin{array}{rcl} u_{11}x_1 + \cdots + u_{1n}x_n & = & 0 \\ u_{21}x_1 + \cdots + u_{2n}x_n & > & 0 \end{array} \right.$$

$$\vdots$$

$$\left\{ \begin{array}{rcl} u_{11}x_1 + \cdots + u_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ u_{s-1\,1}x_1 + \cdots + u_{s-1\,n}x_n & = & 0 \\ u_{s1}x_1 + \cdots + u_{sn}x_n & > & 0 \end{array} \right.$$

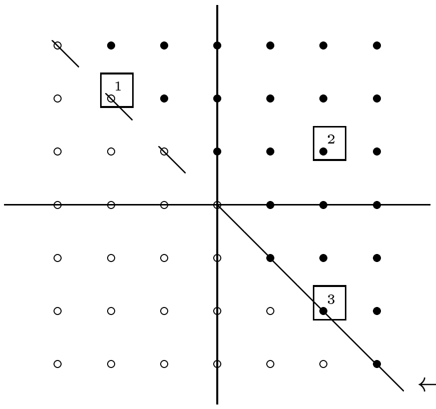
Example 1.62 *The term-ordering* $\tau :=$ deglex *on* $k[x, y]$ *is described by the matrix*

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

and the systems of equations/inequalities associated to τ *are*

$$\{x + y > 0 \quad \textit{and} \quad \left\{ \begin{array}{l} x + y = 0 \\ x > 0 \end{array} \right.$$

In the picture below, the set $P_\tau \subset \mathbb{Z}^2$ *is described by the full dots.*



1) $y^2 \ngeq_\tau x^2 \Rightarrow (-2, 2) \notin P_\tau$

2) $x^2 y >_\tau 1 \Rightarrow (2, 1) \in P_\tau$

3) $x^2 >_\tau y^2 \Rightarrow (2, -2) \in P_\tau$

$\longleftarrow x + y = 0$

It is a natural question exactly which conditions a subset of $\mathbb{Z}^n$ has to satisfy to be the set of positive terms of some term-ordering. To answer, we have to examine the geometry of such sets; we need some geometric definition. The topology is the usual euclidean topology.

Definition 1.63 *A maximal convex subset of* $\mathbb{R}^n$ *not containing the origin is a semi-flag of* $\mathbb{R}^n$.

Proposition 1.64 *Let* $B \subset \mathbb{R}^n$ *be a semi-flag. Then* $B = B_n \cup \ldots \cup B_1$*, where* *a) B_n is an open n-dimensional halfspace of $\mathbb{R}^n$ not containing the origin, whose closure contains the origin;*

b) for each $n-1 \geq i \geq 1$, B_i is an open i-dimensional halfspace of $\mathbb{R}^n$, not containing the origin, contained in the border of B_{i+1}, and whose closure contains the origin.

Proof. The set B is obviously a semi-flag. To show that every semi-flag has the same structure of B, let C be a semi-flag and then take the interior of the closure of C. This is an open n-dimensional halfspace of $\mathbb{R}^n$ not containing the origin, but whose closure contains the origin. We have thus found C_n. The set $C - C_n$ is a semi-flag of $\mathbb{R}^{n-1}$ contained in the border of C, and thus iterating this procedure we find $C_{n-1}, \ldots, C_1$. □

Definition 1.65 *Let B be a semi-flag of $\mathbb{R}^n$; the subset $\mathcal{S}_B := B \cap \mathbb{Z}^n$ is the skeleton of B, and the sets B_i are the components of B.*

An ordering defines a semi-flag and conversely.

Lemma 1.66

a) Let τ be an ordering on T^n. Then there exists a (not necessarily unique) B semi-flag of $\mathbb{R}^n$ whose skeleton is equal to P_τ;

b) if B is a semi-flag of $\mathbb{R}^n$, then there exists an ordering τ on T^n such that $P_\tau = \mathcal{S}_B$.

Proof.

a) Let B be a semi-flag of $\mathbb{R}^n$. Then it is easy to check that the relation defined by

$$\mathbf{x}^{\boldsymbol{a}} > \mathbf{x}^{\boldsymbol{b}} \Leftrightarrow \mathbf{a} - \mathbf{b} \in \mathcal{S}_B$$

is an ordering on T^n.

b) Let $\mathbf{u} := [u_1, \ldots, u_s]$ be the list of vectors in $\mathbb{R}^n$ describing τ, and let the first r, $0 \leq r \leq s$ vectors be in $\mathbb{Q}^n$. Then

$$\left\{ u_{11}x_1 + \cdots + u_{1n}x_n > 0 \right.$$

$$\left\{ \begin{array}{lcl} u_{11}x_1 + \cdots + u_{1n}x_n & = & 0 \\ u_{21}x_1 + \cdots + u_{2n}x_n & > & 0 \end{array} \right.$$

$$\vdots$$

$$\left\{ \begin{array}{lcl} u_{11}x_1 + \cdots + u_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ u_{r-1\,1}x_1 + \cdots + u_{r-1\,n}x_n & = & 0 \\ u_{r1}x_1 + \cdots + u_{rn}x_n & > & 0 \end{array} \right.$$

describe the subsets of $\mathbb{R}^n$ $B_n, B_{n-1}, \ldots, B_{n-r+1}$

Let B_j, $n-r+1 \leq j \leq n-s+1$ be the subsets of $\mathbb{R}^n$ described by

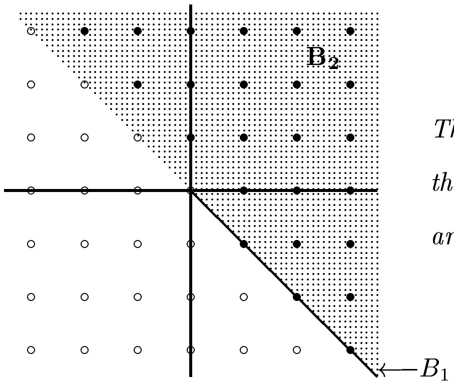
$$\left\{ \begin{array}{lcl} u_{11}x_1 + \cdots + u_{1n}x_n & = & 0 \\ \vdots & \vdots & \vdots \\ u_{j-1\,1}x_1 + \cdots + u_{j-1\,n}x_n & = & 0 \\ u_{j1}x_1 + \cdots + u_{jn}x_n & > & 0 \end{array} \right.$$

It is easy to check that $B_n, \ldots, B_{n-s+1}$ are the first s components of a semi-flag. There exists many semi-flags which have first components $B_n, \ldots, B_{n-s+1}$, and every one of these satisfies the thesis. By the definition of $B_n \ldots B_{n-s+1}$ the skeleton of B is equal to P_τ (cfr. remark 1.61). □

Remark 1.67 *From the proof of the lemma 1.66, it follows that for rational orderings, the associated semi-flag is unique. For more general term-orderings, this is not true.*

Example 1.68 *The semi-flag B associated to the term-ordering $\tau :=$ deglex on $k[x,y]$ it is defined by*

$$B_2 := \{(x,y) \in \mathbb{R}^2 \mid x+y>0\} \cup B_1 := \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; \begin{cases} x+y=0 \\ x>0 \end{cases}\right\}$$



The full dots represent the skeleton of B

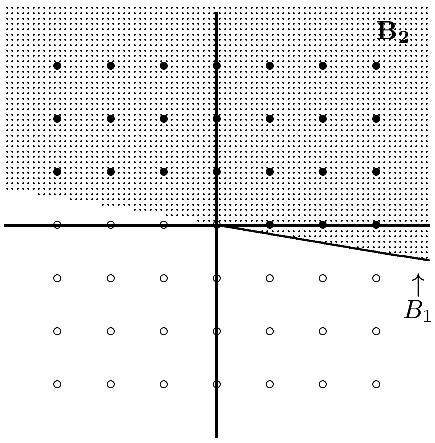
the shadowed area B_2

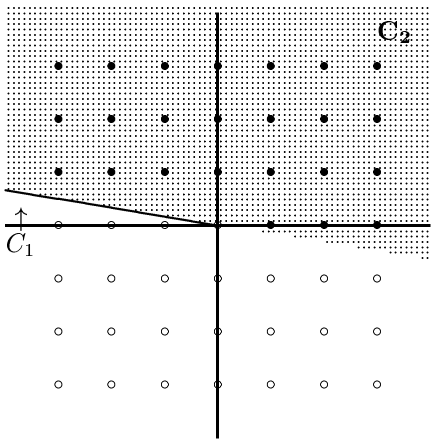
and the halfline from the origin B_1

Example 1.69 *Consider the term-ordering τ described by the vector list* $\mathbf{u} := [(1, \sqrt{37})]$ *introduced in the example 1.49. Two of the associated semi-flags are B and C, whose components are, respectively,*

$$B_2 := \{(x,y) \in \mathbb{R}^2 \mid \{x+\sqrt{37}y>0\}\} \qquad C_2 := \{(x,y) \in \mathbb{R}^2 \mid \{x+\sqrt{37}y>0\}\}$$

$$B_1 := \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; \begin{cases} x+\sqrt{37}y=0 \\ x>0 \end{cases}\right\} \quad \textit{and} \quad C_1 := \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; \begin{cases} x+\sqrt{37}y=0 \\ y>0 \end{cases}\right\}$$





Note that $\mathcal{S}_B \cap B_1 = \emptyset = \mathcal{S}_C \cap C_1$

1.2.3 Finding orderings

Problem 1.70 *Let $R := k[x_1, \dots, x_n]$, $\{f_1, \dots, f_k\} \subset R$ and $t_i \in \mathrm{Supp}\,(f_i)$, $i : 1 \dots k$. We want to find if there exists a term-ordering τ such that $\mathrm{Lt}_\tau(f_i) = t_i$.*

Proposition-Definition 1.71 *Let $R := k[x_1, \dots, x_n]$, $\tau, \sigma \in \mathrm{TO}(R)$ and $F := \{f_1, \dots, f_k\} \subset R$. Then τ is* equivalent *to σ with respect to F ($\tau \sim_F \sigma$) if $\mathrm{Lt}_\tau(F) = \mathrm{Lt}_\sigma(F)$. The relation $\sim_F$ is an equivalence relation.*

Proof. Easy check. □

Definition 1.72 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and $F := \{f_1, \dots, f_k\} \subset R$. The the set of all half-lines associated to a term-ordering equivalent to τ is $C_\tau(F) \subset (\mathbb{R}^n)^+$. If $\underline{\sigma}$ is an equivalence class with respect to $\sim_F$, $C_{\underline{\sigma}}(F)$ is well defined.*

Remark 1.73 *Let $R := k[x_1, \dots, x_n]$, $\{f_1, \dots, f_k\} \subset R$ and $t_i \in \mathrm{Supp}\,(f_i)$, $i : 1 \dots k$. Then the term-orderings τ for which $\mathrm{Lt}_\tau(F) = t_i$ are the term-orderings τ solutions of the system*

$$\left\{ \; t_i >_\tau m \quad m \in \mathrm{Supp}\,(f_i) - t_i, \; i : 1 \dots k \right. \tag{1.4}$$

Proposition 1.74 *With the notation of the remark above, a term-ordering τ is a solution of system 1.4 if and only if its associated matrix $\mathbf{u} := (u_{i,j})$ is a non trivial solution of the list of the systems of linear equation/inequalities (in the $u_{i,j}$ variables), in the sense that it has to fulfill at least one of the systems below.*

$$\left\{ \; u_1 \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) > 0 \quad m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \right. \tag{1.5}$$

$$\left\{ \begin{array}{ll} u_1 \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) = 0 & m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \\ u_2 \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) > 0 & m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \end{array} \right. \tag{1.6}$$

$$\vdots \quad \vdots$$

$$\left\{ \begin{array}{ll} u_1 \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) = 0 & m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \\ \vdots & \vdots \\ u_{n-1} \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) = 0 & m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \\ u_n \cdot (\log(\mathrm{Lt}_\sigma(f_i)) - \log(m)) > 0 & m \in \mathrm{Supp}\,(f_i) i : 1 \dots k \end{array} \right. \tag{1.7}$$

Proof. This is only a rewriting of theorem 1.43. □

Corollary 1.75 *The interior, with respect to the usual euclidean topology, of the union of the set of the solution of the systems in proposition 1.74 is a non-void polyhedral cone in $(\mathbb{R}^n)^+$ with vertex in the origin.*

Proof. The solutions of system (1.5) are the intersection of half-spaces comprising the origin, hence the interior of the subset of $(\mathbb{R}^n)^+$ they make is a non-void polyhedral cone in $(\mathbb{R}^n)^+$ with vertex in the origin. The solution of the other systems are on the boundary of such cone. □

Corollary 1.76 *1. In $C_\sigma(F)$ there exists a vector belonging to $(\mathbb{N}^n)^+$. hence, in every equivalence class with respect to F there is a term-ordering whose first vector belongs to $(\mathbb{N}^n)^+$.*

2. *It is possible to find such vector by finding the rational solutions of the system of linear inequalities 1.5 and then multiplying the solution by a suitable integer.*

Proof. Follows immediately from the discussion above. □

To solve a system of linear inequalities on the rationals, a quite efficient method is the simplex algorithm. It gives a rational solution of the system if such solution exists, or gives notice that the system has no solution. While more complex convex geometry methods are available, namely the Newton polyhedra methods [GS], the simplex algorithms is usually fast enough and very simple to implement [Ro]. Moreover, on the simplex algorithm there exist a huge licterature.

Problem 1.77 *Let $R := k[x_1, \dots, x_n]$, $\{f_1, \dots, f_k\} \subset R$ and $t_i \in \operatorname{Supp}(f_i)$, $i : 1 \dots k$. We want to find all the term-orderings τ for which $\operatorname{Lt}_\tau(f_i) = t_i$ which satisfy some constraint such "they are degree compatible" or "they are block orderings for the first r variables" and so on. The solution is simply to add the constraint to the system 1.5 and solve it. For reasonable constraints, and namely for the two cited before, the results of corollary 1.76 hold. See the example below.*

Example 1.78 *Let $R := k[x, y, z, t]$ and $F := x^2 + xy + y^2 + t$. Let P be the set of potential leading terms of f.*

$\operatorname{Lt}_{\operatorname{lex}(x>..)}(f) = x^2$, *hence* $x^2 \in P$.

$\operatorname{Lt}_{\operatorname{degrevlex}(y>..)}(f) = y^2$, *hence* $y^2 \in P$.

$\operatorname{Lt}_{\operatorname{lex}(t>..)}(f) = t$, *hence* $t \in P$.

If $xy \in P$ then $xy >_\tau x^2$ and $xy >_\tau y^2$ for some $\tau \in \operatorname{TO}(R)$. But then $x >_\tau y$ and $y >_\tau x$, a contradiction. Thus $xy \notin P$. If we restrict our attention to degree-compatible term-orderings, then $t \notin P$.

1.3 Module implementation and orderings

1.3.1 Module implementation

Before the description of the classification for the module orderings compatible with the operations and which give the same ordering to every component, it is useful to examine the exact way in which modules are implemented as data structures.

There are two canonical ways of implementing an $R := k[x_1, \dots, x_n]$ module $M \subset R^r$:

i) the matrix representation, the elements are vectors of R^r; or

ii) the polynomial representation, the elements are polynomials of a suitable polynomial ring $S \supset R$ in which R is embedded.

Proposition 1.79 *Let $R := k[x_1, \dots, x_n]$, $E_1, \dots, E_v$ be new variables and $t_1, \dots, t_r$ terms of $k[\mathbf{x}, \mathbf{E}]$ linearly independents over $k[\mathbf{x}]$. Then the module morphism*

$$\begin{array}{cccc} \phi : & R^r & \longrightarrow & k[\mathbf{x}, \mathbf{E}] \\ & \mathbf{e}_i & \rightsquigarrow & t_i \end{array}$$

is injective

Proof. Easy check. □

Remark 1.80 *Without loss of generality it is possible to have $\mathbf{E} = E$ and $\{t_1, \dots, t_r\} = \{E, E^2, \dots, E^r\}$. In this case, R^r may be embedded in $k[\mathbf{x}, E]/(E^r)$, and ϕ is a module isomorphism. If nothing is said to the contraire, we are supposed to be in this situation.*

Remark 1.81 *The submodules of R^r are embedded in $k[\mathbf{x},\mathbf{E}]$ by the same ϕ the embeds R^r.*

To operate in $\phi(M)$, some definition is needed:

Definition 1.82 *Let M be a $R := k[x_1,\dots,x_n]$-module and $\mathbf{e}_i$ the unitary vectors. Then the set $\{t\cdot\mathbf{e}_i \mid t \in T^n\}$ is the set of the module terms of M and is denoted as $MT(M)$.*

Definition 1.83 *Let $R := k[\mathbf{x},\mathbf{E}]$. Then the variables $E_1,\dots,E_v$ are* module variables*; if a term $x_1^{a_1}\cdots x_n^{a_n}E_1^{b_1}\cdots E_r^{b_r} \in k[\mathbf{x},\mathbf{E}]$ divides a term $x_1^{a'_1}\cdots x_n^{a'_n}E_1^{b'_1}\cdots E_r^{b'_r} \in k[\mathbf{x},\mathbf{E}]$* as module term *if $x_1^{a_1}\cdots x_n^{a_n} \mid x_1^{a'_1}\cdots x_n^{a'_n}$ and $E_1^{b_1}\cdots E_r^{b_r} = E_1^{b'_1}\cdots E_r^{b'_r}$;*

Then, we work not in R^r, but in its isomorphic image in $k[\mathbf{x},\mathbf{E}]$. The module elements are translated as polynomials, and we are to investigate to which extent it is possible to work with the translation of submodules of R^r as ideals.

The main advantage of the representation through the embedding are

a) the homogeneity of the data structure, since systems of generators for both ideals and modules are given by a list of polynomial.

b) the representation of the module elements is naturally sparse.

The main drawback is that, at the implementation level, the additional variables uses more space. Moreover, since polynomials and module elements are treated in the same way, optimizations for the different cases are not natural. The representation through embedding has been used in the PoSSo system. The opposite approach (consider ideals as a special case of modules) has been used in the CoCoA system [CN]. Efficiency comparison between the approaches is quite difficult, since it has to be done in the same system and with the same optimization level. No thorough comparison has been done, but from the author experience with the CoCoA system it seems that the embedding approach leads to easier implementations but is slightly slower, at least without extensive optimization.

Graduations are easily translated by the embedding.

Proposition 1.84 *In the hypothesis of proposition 1.79, the morphism ϕ is graded of degree 0 if the module variables $\mathbf{E}$ have degree 0. If the module R^r has shifts $(d_1,\dots,d_n)$, then the morphism ϕ is graded of degree 0 if the module variables $\mathbf{E}$ have degree such that the terms $t_1,\dots,t_r$ have degrees $d_1,\dots,d_r$. If we want to keep the degree of the module variables standard, we have to introduce additional variables.*

Proof. Easy check. □

Example 1.85 *Let $M := k[x,y](-2)\oplus k[x,y](-3)$ be a $k[x,y]$ module graded by the standard graduation. Then the morphism*

$$\begin{array}{ccc} M & @>\phi>> & k[x,y,E_1,E_2] \\ e_i & \rightsquigarrow & E_i \end{array}$$

is graded of degree 0 if on $k[x,y,E_1,E_2]$ we put system of weights $(1,1,2,3)$. If we want the standard graduation of the overring, we use

$$\begin{array}{ccc} M & @>\psi>> & k[x,y,E_1,E_2,d_1,d_2] \\ e_1 & \rightsquigarrow & E_1d_1^2 \\ e_2 & \rightsquigarrow & E_2d_1^3 \end{array}$$

The matrix notation may be quite handy for a module.
Let $M := (< x,y,1 >, < x^2+1,0,3 >, < 0,xyz,1 >) \subset k[x,y,z]^3$ with the matrix notation M is represented as

$$\begin{bmatrix} x & y & 1 \\ x^2+1 & 0 & 3 \\ 0 & xyz & 1 \end{bmatrix}$$

1.3.2 Module orderings

Since the algorithms that we are interested in are strongly dependent from the ordering, to be useful the embedding has to be ordered. The proof that for every compatible module ordering which give the same ordering to every component there exists a suitable embedding such that ϕ is ordered is the last part of this chapter. From this proof follows the classification of the such module term-orderings (theorem 1.88). The operations we are interested in, and their module interpretations/optimizations, use these orderings. More general module orderings are encountered, especially in problems coming from differential algebra and differential equation solving. In this setting the classification problem has been studied for Riquier orderings by Riquier [Ri], Thomas [Th] and others. More recently, Carrà-Ferro and Sit [CSi] have studied more general, non-Riquier module orderings. Reid and Rust [Ru, Ru2, RR] have classified *all* module orderings, after proving that there exists more module orderings that the ones considered by Carrà-Ferro and Sit. For a complete discussion on the literature, see [Ru].

Definition 1.86 *Let M be a free R module, $\{\mathbf{e}_i\}_{i\in S}$ its unitary vectors and $\tau \in \mathrm{TO}(T^n)$ then*

1. *the set $MT(M) := \{t\,\mathbf{e}_i \mid t \in T^n\}$ is the set of the module-terms of M;*
2. *let $a,\, b \in T^n$ and $m,\, n \in MT(M)$. Then, a* Riquier ordering *μ is a total ordering on $MT(M)$such that*

$$a >_\tau b \Rightarrow am >_\mu bm \;, m >_\mu n \Rightarrow am >_\mu an \;\text{ and } m\,\mathbf{e}_i > n\,\mathbf{e}_i \Rightarrow m\,\mathbf{e}_j > n\,\mathbf{e}_j \;\forall\; j \in S$$

If $a\,\mathbf{e}_1 > \mathbf{e}_1 \;\forall\; a \in T^n$, then μ is a Riquier term-ordering.

Some examples of Riquier ordering are given in example 1.95-1.97.

Example 1.87 *Let $R := k[x]^2$ and μ the total module ordering on $MT(R)$ defined by*

$$x^a\,\mathbf{e}_i >_\mu x^b\,\mathbf{e}_j \Leftrightarrow \begin{cases} i = j = 1 \text{ and } a > b \\ i = j = 2 \text{ and } a < b \\ i > j \end{cases}$$

It is easy to see that μ is a total ordering on R^2, but μ is not a Riquier ordering, since $x\,\mathbf{e}_1 >_\mu 1\,\mathbf{e}_1$ but $x\,\mathbf{e}_2 <_\mu 1\,\mathbf{e}_2$.

Theorem 1.88 (Classification theorem) *Let τ be an ordering on T^n and M be a free, finitely generated R-module of rank r. Let μ be a Riquier module ordering on $MT(M)$. Then there are k, $(k \geq 1)$, variables $y_1, \ldots, y_k$, an ordering μ'on $k[\mathbf{x},\mathbf{y}]$ and $t_1, \ldots, t_r$ terms of $k[\mathbf{x},\mathbf{y}]$ linearly independent on $k[\mathbf{x}]$ such that*

1. *the module morphism*

$$\begin{array}{ccc} (M,\mu) & @>\phi>> & (k[\mathbf{x},\mathbf{y}],\mu') \\ \mathbf{e}_i & \rightsquigarrow & t_i \end{array}$$

is ordered and injective.

2. *μ is the restriction of μ' to $\phi(M)$.*

Proof. Since ϕ is clearly injective, M is isomorphic to $\phi(M)$. To prove the theorem, it suffices to define a semi-flag B of $\mathbb{R}^{n+k}$ such that ϕ is ordered with respect to μ and the ordering μ' described by the skeleton of B on T^{n+k}.

Let t_i be $x_1^{\alpha_{i,1}}, \dots, x_n^{\alpha_{i,n}}$ for each $1 \le i \le r$. We associate to every module-term in M a vector in $\mathbb{R}^{n+k}$ in a natural way

$$\mathbf{x}^{\boldsymbol{a}}\mathbf{e}_i \rightsquigarrow P_{a,i} := (a_1 + \alpha_{i,1}, \dots, a_n + \alpha_{i,n}, 0, \dots, \underbrace{1}_{n+i}, \dots, 0)$$

Let us consider the subset $\triangle$ of $\mathbb{Z}^n \times \mathbb{Z}^r$ defined by

$$\triangle := \{P_{a,i} - P_{b,j} \mid \mathbf{x}^{\boldsymbol{a}}\mathbf{e}_i > \mathbf{x}^{\boldsymbol{b}}\mathbf{e}_j\}$$

note that there are two kinds of the elements in $\triangle$:

$(a_1 - b_1, \dots, a_n - b_n, 0, \dots, 0)$ given by$P_{\mathbf{a},i} - P_{\mathbf{b},i}$

$(a_1 - b_1, \dots, a_n - b_n, 0, \dots, \underbrace{1}_{n+i}, \dots, \underbrace{-1}_{n+j}, \dots, 0)$ given by $P_{\boldsymbol{a},i} - P_{\boldsymbol{b},j}$.

Note also that $0 \notin \triangle$.

Claim 1.89 *There exists a convex subset $C \subset \mathbb{R}^{n+k}$ which contains $\triangle$ but not the origin.*

Since a semi-flag of $\mathbb{R}^{n+k}$ is a maximal convex subset of $\mathbb{R}^{n+k}$ which does not contain the origin, there exists a semi-flag B which contains C, and hence $\triangle$. The skeleton of B defines a term-ordering μ'. Since the skeleton of B contains $\triangle$, we have that

$$\begin{aligned} \mathbf{x}^{\boldsymbol{a}}\mathbf{e}_i >_\mu \mathbf{x}^{\boldsymbol{b}}\mathbf{e}_j &\Leftrightarrow (a_1 - b_1, \dots, a_n - b_n, 0, \dots, \underbrace{1}_{n+i}, \dots, \underbrace{-1}_{n+j}, \dots, 0) \in \triangle \\ &\Leftrightarrow \phi(\mathbf{x}^{\boldsymbol{a}}\mathbf{e}_i) = \mathbf{x}^{\boldsymbol{a}} y_i >_{\mu'} \mathbf{x}^{\boldsymbol{b}} y_j = \phi(\mathbf{x}^{\boldsymbol{b}}\mathbf{e}_j) \end{aligned}$$

Hence ϕ is ordered with respect to μ and μ'.

Proof *of the claim.*

Let $C := \{\sum_i c_i P_i \mid P_i \in \triangle \text{ and } \sum_i c_i = 1\}$, e. g. the convex hull of $\triangle$. We have that $\triangle \subset C \subset \mathbb{R}^{n+k}$. Then we have only to prove that C does not contain the origin. Looking for a contradiction, we suppose that there exist $\delta_1, \dots \delta_v \in \triangle$ such that

$$\sum_{i=1}^{v} c_i \delta_i = 0 \tag{1.8}$$

where the $\sum_i c_i \ge 0$. If the set of c_i's which satisfy (1.8) has real points, then it has rational (and hence integer) points. Thus disproving (1.8) for integer c_i's is enough.
We then consider the list $L := [\underbrace{\delta_1, \dots, \delta_1}_{c_1}, \dots, \underbrace{\delta_v, \dots, \delta_v}_{c_v}]$ and rewrite equation (1.8) as

$$\sum_{\delta \in L} \delta = 0 \tag{1.9}$$

Either all the elements of L have the last r components null, or they don't. If they don't, there exists in L an element δ whose 1 i^{th}, $n+1 \le i \le n+r$ component is 1. By equation (1.9) there exists δ' in L whose i^{th} component is -1. Then let $L' := L - \{\delta, \delta'\} \cup \{\delta + \delta'\}$. The vector $\delta + \delta'$ belong to $\triangle$ since μ is compatible with the monoid operation, and the sum $\sum_{\delta \in L'}$ is still equal to zero. Since every application of the above process to L gives a list with a lesser number of elements, after a finite number of iterations all the elements of L are in $\triangle$ and have the last r components null. Thus, every element $\delta_i = (c_1^i, , c_n^i, 0, \dots, 0)$ of L comes from an inequality $\mathbf{x}^{\boldsymbol{a}_i}\mathbf{e}_{l_i} > \mathbf{x}^{\boldsymbol{b}_i}\mathbf{e}_{l_i}$, where $\mathbf{a}_i$ is the vector of the positive

c_i^j and $\mathbf{b}_i$ is the vector of the negative c_i^j. Since μ is a Riquier ordering, we may suppose that $l_i = 1\ \forall i$. That implies

$$\mathbf{x}^{\sum_i \mathbf{a}_i} > \mathbf{x}^{\sum_i \mathbf{b}_i} \Rightarrow \sum_i (\mathbf{a}_i - \mathbf{b}_i) \neq 0 \Longrightarrow \sum_{\delta \in L} \delta \neq 0$$

a contradiction. $\square$

Remark 1.90 *From theorem 1.88, the classification of Riquier orderings derives easily from the classification of ring orderings.*

Remark 1.91 *The obvious representation of an element of M as n-tuple of elements of $k[\mathbf{x}]$ is not convenient from the efficiency point of view, unless the term-ordering is "lexicographic on the components", e. g. $t\mathbf{e}_i > m\mathbf{e}_j$ if $i < j$. Only in this case, if the polynomials are represented as usual as sum of monomials in decreasing order, the leading term is the first non zero term. The representation of module elements as polynomials thorough ϕ and μ' should more efficient in the general case. The implementations of polynomial data structure and arithmetics, and the relative optimization of the polynomial with respect to module operations are crucial to this matter.*

1.3.3 Some interesting module term-orderings classes

From the point of view of computational commutative algebra, there are two classes of Riquier term-orderings which are especially significant, the Riquier term-orderings based on a ring term-ordering plus position considerations (cfr. example 1.95) and the "weighted" Riquier term-orderings associated to syzygies and resolutions computations [C, MM] (cfr. example 1.96-1.97).

Definition 1.92 *Let M be a free R-module of rank r and $\tau \in \mathrm{TO}(T^n)$. A Riquier module term-ordering μ on M is a*

To+Pos$^+$ term-ordering if $t\,\mathbf{e}_i >_\mu t'\mathbf{e}_j \Longleftrightarrow t >_\tau t'$ or $t = t'$ and $i > j$;

To+Pos$^-$ term-ordering if $t\,\mathbf{e}_i >_\mu t'\mathbf{e}_j \Longleftrightarrow t >_\tau t'$ or $t = t'$ and $i < j$;

Pos+To$^+$ term-ordering if $t\,\mathbf{e}_i >_\mu t'\mathbf{e}_j \Longleftrightarrow i > j$ or $i = j$ and $t >_\tau t'$;

Pos+To$^-$ term-ordering if $t\,\mathbf{e}_i >_\mu t'\mathbf{e}_j \Longleftrightarrow i < j$ or $i = j$ and $t >_\tau t'$.

Definition 1.93 *Let M be a free R-module of rank r, $\tau \in \mathrm{TO}(T^n)$ and $t_1, \dots, t_r$ be terms in T^n. A Riquier module term-ordering μ on M is a* weighted term-ordering *with respect to the terms $m_1, \dots, m_r$ in T^n if*

$$t\,\mathbf{e}_i >_\mu t'\,\mathbf{e}_j \Longrightarrow tm_i >_\tau t'm_j \text{ or } tm_i = t'm_j \text{ and } i > j$$

The matrices associated to the ring term-ordering τ and $lex(x_1 > \cdots > x_r)$ will be denoted by Γ_τ and Γ_{lex_r} respectively.

Remark 1.94 *Using the notation of the classification theorem, it is easy to see that given a free, finitely generated R-module M, there is a considerable degree of freedom in the choice of the overring and of the embedding associated to a given module term-ordering. In the following we exhibit two different choices of overring (with term-ordering and associated matrix) and embedding for several examples. Clearly, the smaller the number of new variables introduced, the more efficient is the handling of the ordering.*

Example 1.95 *With the notation introduced above, A possible choice of overring and embedding for the To+Pos$^+$ module term-ordering is*

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x}, \mathbf{y}] \\ & \mathbf{e}_i & \rightsquigarrow & y_i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} [\Gamma_\tau] & \mathbf{0} \\ \mathbf{0} & [\Gamma_{lex_r}] \end{bmatrix}$$

Another choice is

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x}, y] \\ & \mathbf{e}_i & \rightsquigarrow & y^i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} \begin{bmatrix} & \Gamma_\tau & \end{bmatrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 0 \quad \cdots \quad 0 & 1 \end{bmatrix}$$

Example 1.96 *With the notation introduced above, A possible choice of overring and embedding for the weighted (with respect to the terms $m_1, \ldots, m_r$ in T^n) module term-ordering is*

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x}, \mathbf{y}] \\ & \mathbf{e}_i & \rightsquigarrow & y_i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} \begin{bmatrix} & \Gamma_\tau & \end{bmatrix} & \begin{bmatrix} u_1 \cdot Log(m_1) & \cdots & u_1 \cdot Log(m_r) \\ \vdots & \vdots & \vdots \\ u_n \cdot Log(m_1) & \cdots & u_n \cdot Log(m_r) \end{bmatrix} \\ \begin{bmatrix} & \mathbf{0} & \end{bmatrix} & \begin{bmatrix} & \Gamma_{lex_r} & \end{bmatrix} \end{bmatrix}$$

where $\Gamma_\tau := (u_1, \ldots, u_n)$.
If the number of variables $\mathbf{x}$ is n, another choice is

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x}, \mathbf{y}] \\ & \mathbf{e}_i & \rightsquigarrow & m_i y^{n-i+1} \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} \begin{bmatrix} & \Gamma_\tau & \end{bmatrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 0 \quad \cdots \quad 0 & -1 \end{bmatrix}$$

Note that μ' is not a term-ordering on $k[\mathbf{x}, \mathbf{y}]$, but its restriction to $\phi(M)$ is.

Let us examine in detail a weighted term-ordering.

Example 1.97 *Let M be the module $(k[x_1, x_2])^3$. The term-ordering τ on $k[\mathbf{x}]$ is* $\mathrm{deglex}(x_1 > x_2)$ *and the compatible module term-ordering μ on M is the multiweigthed module term-ordering with respect to the terms x_1^5, $x_1^4 x_2$, $x_1^3 x_2^2$. A possible choice of overring and embedding is*

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[x_1, x_2, y_1, y_2, y_3] \\ & \mathbf{e}_i & \rightsquigarrow & y_i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} 1 & 1 & 5 & 5 & 5 \\ 1 & 0 & 5 & 4 & 3 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Another choice of overring and embedding is

$$\begin{array}{cccc} \psi: & M & \longrightarrow & k[x_1, x_2, y] \\ & \mathbf{e}_1 & \rightsquigarrow & x_1^3 x_2^2 y \\ & \mathbf{e}_2 & \rightsquigarrow & x_1^4 x_2 y^2 \\ & \mathbf{e}_3 & \rightsquigarrow & x_1^5 y^3 \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Such orderings are in fact ring block orderings on the overring.
Hereafter, all the ring term-orderings considered will be rational; the matrices associated to the ring term-ordering τ will be denoted by Γ_τ.

Remark 1.98 *Using the notation of the classification theorem, it is easy to see that given a free, finitely generated R-module M, there is a considerable degree of freedom in the choice of the overring and of the embedding associated to a given module term-ordering. In the following we exhibit two different choices of overring (with term-ordering and associated matrix) and embedding for several examples.*

Example 1.99 *Let* lex *be the lexicographic term-orderingon $k[\mathbf{y}]$. With the notation introduced in the theorem 1.88, a possible choice of overring and embedding for the To+Pos$^+$ module term-ordering is*

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x},\mathbf{y}] \\ & \mathbf{e}_i & \rightsquigarrow & y_i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} [\Gamma_\tau] & \mathbf{0} \\ \mathbf{0} & [\Gamma_{lex}] \end{bmatrix}$$

and $\mu' = \tau \times \mathrm{lex}$. Another choice is

$$\begin{array}{cccc} \phi: & M & \longrightarrow & k[\mathbf{x},y] \\ & \mathbf{e}_i & \rightsquigarrow & y^i \end{array} \qquad \Gamma_{\mu'} := \begin{bmatrix} \begin{bmatrix} & & \\ & \Gamma_\tau & \\ & & \end{bmatrix} & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \begin{matrix} 0 & \cdots & 0 \end{matrix} & -1 \end{bmatrix}$$

and $\mu' = \tau \times \mathrm{lex}$ Note that μ' is not a term-ordering on $k[\mathbf{x},\mathbf{y}]$, but its restriction to $\psi(M)$ is a term-ordering.

There are orderings on the overring that are not induced by a Riquier ordering of M via the embedding, like the following

Example 1.100 *Let $M := (k[x,y])^2$, τ and μ be as in example. Then the term-ordering μ' on M given by the matrix*

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

is not compatible with μ through ψ, since $x_2\,\mathbf{e}_3 >_\mu x_1\,\mathbf{e}_1$ but $\psi(x_2\,\mathbf{e}_3) = x_1^5 x_2 y^3 <_{\mu'} x_1^4 x_2^2 y = \psi(x_1\,\mathbf{e}_1)$.

1.4 Hilbert Functions and Poincaré series

In this section we give a very short description of some features of Hilbert function theory which are useful for the optimization of the operations we are interested with. Some further details are shown in chapter 2, when we discuss the use of the Hilbert function for the optimization of the Buchberger algorithm.

Definition 1.101 *Let $\mathbf{w}$ be a system of weights, $R := k[x_1, \ldots, x_n]$ be a $\mathbf{w}$ graded ring and M be a $\mathbf{w}$ graded R module. The function*

$$\begin{array}{cccc} \mathcal{H}_M^{\mathbf{w}}: & \mathbb{N}^m & \longrightarrow & \mathbb{N} \\ & \mathbf{d} & \rightsquigarrow & dim_k(M_{\mathbf{d}}) \end{array}$$

is the Hilbert function *of M with respect to $\mathbf{w}$.*

Definition 1.102 *With the notation of the preceding definition, the series*

$$\mathcal{HP}_M^{\mathbf{w}}(\lambda) := \sum_{d \in \mathbb{N}^m} \mathcal{H}_M^{\mathbf{w}}(\mathbf{d})\lambda^{\mathbf{d}} \in \mathbb{Z}[[\lambda_1, \ldots, \lambda_m]]$$

is the Poincaré series *of M with respect to $\mathbf{w}$.*

When there is no danger of ambiguity, we write $\mathcal{H}_M$ instead of $\mathcal{H}_M^{\mathbf{w}}$ and $\mathcal{HP}_M$ instead of $\mathcal{HP}_M^{\mathbf{w}}$.

The following lemma collect several basic facts on Poincaré series.

Lemma 1.103 *Let* $\mathbf{w}$ *be a system of weights* $R := k[x_1, \ldots, x_n]$ *a* $\mathbf{w}$ *graded ring, I a* $\mathbf{w}$ *homogeneous ideal of* R *and* $\sigma \in \mathrm{TO}(T^n)$*. Then*

Macaulay Theorem $\mathcal{HP}^{\mathbf{w}}_{R/I} = \mathcal{HP}^{\mathbf{w}}_{R/\mathrm{Lt}_\sigma(I)}$

Hilbert-Serre Theorem *There exists* $f \in \mathbb{Z}[\lambda_1, \ldots, \lambda_m]$ *s.t.* $\mathcal{HP}^{\mathbf{w}}_{R/I}(\lambda_1, \ldots, \lambda_m) = \frac{f(\lambda_1,\ldots,\lambda_m)}{\prod_{i=1}^n (1-\lambda^{\partial_{\mathbf{w}}(x_i)})}$

Ring formula *[Euler]* $\mathcal{HP}^{\mathbf{w}}_{R}(\lambda) = \prod_{i=1}^n \frac{1}{(1-\lambda^{\partial_{\mathbf{w}}(x_i)})}$

Proof. Macaulay and Hilbert-Serre theorems are classical results. It is well known that the coefficient of the term of $\mathbf{w}$ degree $\boldsymbol{\beta}$ in the series inverse of $\prod_{i=1}^n \frac{1}{(1-\lambda^{\partial_{\mathbf{w}}(x_i)})}$ is the number of positive solutions of the system of diofantine equations

$$\{\boldsymbol{\alpha}_1 \partial_{\mathbf{w}}(x_1) + \ldots + \boldsymbol{\alpha}_n \partial_{\mathbf{w}}(x_n) = \boldsymbol{\beta}$$

The proof of the ring formula comes easily from that fact. □

Lemma 1.104 *Let* $\mathbf{w}$ *be a system of weights,* $R := k[x_1, \ldots, x_n]$ *be a* $\mathbf{w}$ *graded ring,,* $M \subset R^r$ *be a* $\mathbf{w}$ *graded R module and* $\tau \in \mathrm{TO}(M)$*. Then, if* $(Lt_\tau(M))_i$ *are the leading terms of M which lies in the i-th component, then*

$$\mathcal{HP}^{\mathbf{w}}_{M}(\lambda) := \sum_{i:=1}^{r} \mathcal{HP}^{\mathbf{w}}_{(Lt_\tau(M))_i}(\lambda)$$

Hence, if M has rank r, then

$$\mathcal{HP}^{\mathbf{w}}_{R^r/M}(\lambda) := \sum_{i:=1}^{r} \mathcal{HP}^{\mathbf{w}}_{(R/(Lt_\tau(M))_i)}(\lambda)$$

Moreover, we have

$$\mathcal{HP}^{\mathbf{w}}_{M(-\partial_{\mathbf{w}}(F))}(\lambda) = \lambda^{\partial_{\mathbf{w}}(F)} \mathcal{HP}^{\mathbf{w}}_{M}(\lambda)$$

Proof. Easy check. □

Definition 1.105 *With the notation of lemma 1.103, we denote by* $\langle I \rangle^{\mathbf{w}}$ *or simply by* $\langle I \rangle$ *the polynomial* $f(\lambda_1, \ldots, \lambda_m)$ *and we call it the* $\mathcal{HP}$-numerator *of I with respect to* $\mathbf{w}$*. If M is an R module the definition of* $\langle M \rangle$ *is analogous to the definition of* $\langle I \rangle$*.*

Remark 1.106 *The computation of* $\mathcal{H}^{\mathbf{w}}_{R/I}$ *from* $\mathcal{HP}^{\mathbf{w}}_{R/I} = \frac{f(\lambda_1,\ldots,\lambda_m)}{\prod_{i=1}^n (1-\lambda^{\partial_{\mathbf{w}}(x_i)})}$ *is quite easy. It suffices to invert the series to the denominator and then multiply the result by f to get the canonical representation of the Poincaré series from which it is immediate to read the Hilbert function. In the module case, by lemma 1.104, it suffices to sum the Hilbert functions of the leading term ideals of the components. For type one graduations, there are handy closed formulas, see e. g. [R3]. For higher type graduations, the things are a bit more complex.*

Proposition 1.107 *Let* $0@ >>> M@ > f >> N@ > g >> Q@ >>> 0$ *be a graded short exact sequence of A modules of degree* 0*. Then the functor*

$$\dim{}_k(\cdot) : \{A \text{ modules free, f.g.}\} \longrightarrow \mathbb{N}$$

is additive on the sequence (e. g. $\dim{}_k(N) = \dim{}_k(M) + \dim{}_k(P)$*).*

Proof. Easy check. □

Corollary 1.108 *Let M, N, P be R free, finitely generated R modules. Then*

1. *If* $0 \to M \xrightarrow{f} N \xrightarrow{g} P \to 0$ *is a graded short exact sequence of degree 0 then*
$$\mathcal{H}_N^{\mathbf{w}}(\mathbf{d}) = \mathcal{H}_M^{\mathbf{w}}(\mathbf{d}) + \mathcal{H}_P^{\mathbf{w}}(\mathbf{d})\ \forall \mathbf{d} \in \mathbb{Z}^n$$
and
$$\mathcal{HP}_N^{\mathbf{w}}(\mathbf{d}) = \mathcal{HP}_M^{\mathbf{w}}(\mathbf{d}) + \mathcal{HP}_P^{\mathbf{w}}(\mathbf{d})\ \forall \mathbf{d} \in \mathbb{Z}^n$$

2. *If M, N are R modules then* $\mathcal{HP}_{M\oplus N}^{\mathbf{w}} = \mathcal{HP}_M^{\mathbf{w}} + \mathcal{HP}_N^{\mathbf{w}}$ $(\mathcal{H}_{M\oplus N}^{\mathbf{w}} = \mathcal{H}_M^{\mathbf{w}} + \mathcal{H}_N^{\mathbf{w}})$.

Proof. Follows from the Hilbert function definition and lemmas 1.103 and 1.104 □

1.4.1 Main formula

Proposition 1.109 *Let $\mathbf{w}$ be a system of weights, $R := k[x_1, \dots, x_n]$ be a $\mathbf{w}$ graded ring, M be a $\mathbf{w}$ graded R module and F be an homogeneous element of M. Then*

a) the sequence
$$0 \to \left(\frac{M}{(0 \underset{M}{:} (F))M}\right)(-\partial_{\mathbf{w}}(F)) \xrightarrow{\cdot F} M \xrightarrow{\pi} \frac{M}{FM} \to 0$$
is a short graded sequence of degree 0.

b)
$$\mathcal{HP}^{\mathbf{w}}_{\left(\frac{M}{FM}\right)}(\lambda) = \mathcal{HP}^{\mathbf{w}}_M(\lambda) - \lambda^{\partial_{\mathbf{w}}(F)} \mathcal{HP}^{\mathbf{w}}_{\left(\frac{M}{(0 \underset{M}{:} (F))M}\right)}(\lambda)$$

Proof. The proof of *a)* is an easy check. The proof of *b)* follows from the proposition 1.108 and *a)*. □

Corollary 1.110 *Let I be an homogeneous ideal of $R := k[x_1, \dots, x_n]$ and $\mathbf{w}$ a system of weights. Then*
$$\mathcal{HP}^{\mathbf{w}}_{R/(I,F)}(\lambda) = \mathcal{HP}^{\mathbf{w}}_{R/I}(\lambda) - \lambda^{\partial_{\mathbf{w}}(F)} \mathcal{HP}^{\mathbf{w}}_{R/(I:(F))}(\lambda)$$
or, using the $\langle\cdot\rangle$ notation,
$$\langle I, F\rangle^{\mathbf{w}} = \langle I\rangle^{\mathbf{w}} + \lambda^{\partial_{\mathbf{w}}(F)} \langle I : (F)\rangle^{\mathbf{w}} \text{[main formula]}$$

Another useful formula comes from the following

Proposition 1.111 *Let $\mathbf{w}$ be a system of weights, $R := k[x_1, \dots, x_n]$, and $M, N \subset R^r$ $\mathbf{w}$ graded R modules. Then the short sequence*
$$0@>>> M \cap N @>>> M \oplus N @>>> M + N @>>> 0$$
is a $\mathbf{w}$ graded short exact sequence of degree 0. Hence
$$\langle M + N\rangle^{\mathbf{w}} = \langle M\rangle^{\mathbf{w}} + \langle N\rangle^{\mathbf{w}} - \langle M \cap (N)\rangle^{\mathbf{w}} \text{ [intersection formula]}$$

Proof. Follows from corollary 1.108 *b)*. □

Proposition 1.112 *Let $R := k[x_1, \dots, x_n]$, $I := (t_1, \dots, t_r)$ be a monomial ideal of R and $t \in T^n$. Let $a, b \in \mathbb{Z}$ and*

$$a \ominus b := \begin{cases} a - b & \text{if } a - b > 0 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} I : (F) &= (t_1, \dots, t_r) : (t)) \\ &= (x_1^{a_1^1} \cdots x_n^{a_n^1}, \dots, x_1^{a_1^r} \cdots x_n^{a_n^r}) : (x_1^{b_1} \cdots x_n^{b_n}) \\ &= (x_1^{a_1^1 \ominus b_1} \cdots x_n^{a_n^1 \ominus b_n}, \dots, x_1^{a_1^r \ominus b_1} \cdots x_n^{a_n^r \ominus b_n}) \end{aligned} \tag{1.10}$$

and

$$\begin{aligned} I \cap (F) &= (t_1, \dots, t_r) \cap (t)) \\ &= (x_1^{a_1^1} \cdots x_n^{a_n^1}, \dots, x_1^{a_1^r} \cdots x_n^{a_n^r}) \cap (x_1^{b_1} \cdots x_n^{b_n}) \\ &= (x_1^{a_1^1 + b_1} \cdots x_n^{a_n^1 + b_n}, \dots, x_1^{a_1^r + b_1} \cdots x_n^{a_n^r + b_n}) \end{aligned} \tag{1.11}$$

Proof. Easy check. □

Remark 1.113 *Using propositions 1.110, 1.111 and 1.112, it is easy to develop algorithms for the effective computation of the $\mathcal{H}$ numerator of a ring. The Poincaré series and Hilbert functions are built very easily once the $\mathcal{H}$ numerator is known. Since it lowers the degrees, the main formula is more useful than the intersection formula. Proposition 1.104 reduces the computation of module Hilbert functions to the computation of ring Hilbert functions.*

1.4.2 M-regular sequences

Definition 1.114 *Let $\mathbf{w}$ be a system of weights, $R := k[x_1, \dots, x_n]$, M be a $\mathbf{w}$ graded R module and $F_1, \dots, F_k$ be homogeneous polynomials of R. Then $(F_1, \dots, F_k)$ is a M-*regular sequence *if $F_i \notin ZD(M/IM)$ for each $i : 1 \dots k$, where I is generated by the first $i-1$ polynomials.*

Proposition 1.115 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights, M be a $\mathbf{w}$ graded R module and $I = (F_1, \dots, F_k) \subseteq R$. Then*

$$\mathcal{HP}_{M/IM}(\lambda) = \mathcal{HP}_M(\lambda) \prod_{i=1}^{k} (1 - \lambda_i^{\partial_{\mathbf{w}}(F_i)})$$

if and only if $(F_1, \dots, F_k)$ are a M regular sequence.

Proof. Follows from the main formula. □

Corollary 1.116 *For a list $F_1, \dots, F_k$ of polynomials the be a regular sequence is independent from the order of the polynomials in the list.*

Regular sequences are important here because for ideals generated by regular sequences the computation of the Hilbert function is quite easy. That is useful for the Hilbert-Driven algorithms, see chapter 2

1.4.3 Some useful formula

From the main formula, we deduce some useful formulas for the computation of Poincaré series.

Remark 1.117 *Let* $\mathbf{w}$ *be a system of weights,* $R := k[x_1, \dots, x_n]$ *be a* $\mathbf{w}$ *graded ring,* I *be an homogeneous ideal of* R *and* $F \in R$ *homogeneous.*

a) $\langle 0 \rangle = 1$;

b) $\langle 1 \rangle = 0$;

c) $\langle F \rangle = 1 - \lambda^{\partial_{\mathbf{w}}(F)}$;

d) $\langle FI \rangle = 1 - \lambda^{\partial_{\mathbf{w}}(F)} + \lambda^{\partial_{\mathbf{w}}(F)} \langle I \rangle$;

e) $\langle I, F \rangle = \langle I \rangle - \lambda^{\partial_{\mathbf{w}}(F)} \langle I : (F) \rangle$.

f) $\langle I \rangle = \langle I, F \rangle + \lambda^{\partial_{\mathbf{w}}(F)} \langle I : (F) \rangle$.

g) $\langle I \rangle = \langle I \rangle + (1 + \lambda^{\partial_{\mathbf{w}}(F)}) + \langle I \cap (F) \rangle$.

Proof. The proofs of *a)-c)* is an easy check. *e)* is the main formula, *f)* is a rewriting of the main formula and *g)* is the intersection formula. Let us prove *d)* by induction on n, the number of generators of I. If $n = 1$, the formula follows easily from the main formula. If $n > 1$, then let $I = (t_1, \dots, t_k)$. From the main formula it follows that $\langle x^a(t_1, \dots, t_k) \rangle = \langle x^a(t_2, \dots, t_k) \rangle - \lambda^{|a|+deg(t_1)} \langle (x^a(t_2, \dots, t_k)) : (x^a t_1) \rangle = 1 - \lambda^{|a|} + \lambda^{|a|} \langle t_2, \dots, t_k \rangle - \lambda^{|a|+deg(t_1)} \langle (t_2, \dots, t_k)) : (t_1) \rangle = 1 - \lambda^{|a|} + \lambda^{|a|} \langle t_1, \dots, t_k \rangle$. □

Definition 1.118 *The polynomial (usually a term)* F *is the* pivot *of the main formula. Different choice strategies for the pivot give different algorithm for the Poincaré series computation.*

Lemma 1.119 *Let* $\mathbf{w}$ *be a system of weights,* $R := k[x_1, \dots, x_n]$ *and* M *be a* $\mathbf{w}$ *graded* R *module. If* $M = m_1, \dots, m_k$ *and the elements* m_i *can be partitioned in the lists* $M_1, \dots, M_p$ *is such a way that the sets of the variables that appear in a list* M_i *are mutually disjoint, then*

$$\langle M \rangle = \prod_{i=1}^{p} \langle M_i \rangle$$

Proof. A consequence of the main formula. □

This lemma allows us a useful simplification of the computation of the Poincaré series series in several easy cases.

Algorithm 1.120 *HF*

Input L list of terms of $k[x_1, \dots, x_n]$, $\mathbf{w}$ system of weights.
Output $\langle (L) \rangle^{\mathbf{w}}$.

begin
  $L :=$Interreduced(L)
  if $L = (t, t')$ **then return** $1 - \lambda^{\partial_{\mathbf{w}}(t)} - \lambda^{\partial_{\mathbf{w}}(t')}(1 - \lambda^{|\partial_{\mathbf{w}}(t) \ominus \partial_{\mathbf{w}}(t')|})$
  if $L = (L', x^{a_1}, \dots, x^{a_k})$, $a_i \in \mathbb{N}$ **then return** $\langle L' \rangle^{\mathbf{w}} (1 - \lambda^{\partial_{\mathbf{w}}(a_1)}) \cdots (1 - \lambda^{\partial_{\mathbf{w}}(a_k)})$
  if $L = (L_1, \dots, L_k)$ with the lists $L_1, \dots, L_k$ s.t. their variable are disjoint
    then return $\langle L_1 \rangle^{\mathbf{w}} \cdots \langle L_k \rangle^{\mathbf{w}}$
  choose $\mathbf{x^a}$ pivot for L
  Compute $J :=$ list of generators of $(L) : (\mathbf{x^a})$
  return $\langle (L, \mathbf{x^a}) \rangle^{\mathbf{w}} + \lambda^{\partial_{\mathbf{w}}(a)} \langle J \rangle^{\mathbf{w}}$
end

The correctness of the algorithm, if it terminates, follows immediately from remark 1.117 and lemma 1.119. The termination is guaranteed if the choice of the pivot is such that $(L, x^{\mathbf{a}})$ is less complex (with respect to the number of genberators, the degree or some other criterion) than L.

Remark 1.121 *This is not he place for a detailed discussion of the algorithms for the computation of the Hilbert Function, for which we refer to [BS, BCR, BCRT, B]. The most efficient algorithms use the main formula with carefully studied pivot choice strategies. It seems that the optimal strategies are different in the unigraded and multigraded cases. The main choices are between the power of a variable and the* GCD *of some terms in the list L. In the first case, the internal machinery in the algorithm are optimized, while in the second case the number of steps is lower. It seems that the variable pivot is best when the system of weights has type one, while for higher type systems of weights the* GCD *pivot is the best choice. Moreover, substantial optimizations are possible in the addition/multiplication giving the result polynomial, especially in the type one case. See [B] for a complete analisys. Here, it suffices to say that the computation of the Hilbert function is usually very fast with respect to the computation of the operations we are interested with, and hence it is possible to use the Hilbert function as a tool during these computations. We remark that to compute a Poincaré series of an ideal with respect to a high type system of weights is much less efficient than compute the Poincaré series of the same ideal with respect to a type one system of weights, see example 2.27.*

The computation of a Poincaré series is usually very fast for examples coming from Gröbner basis computations (e. g. of interest here) and when the type of the system of weights is one. For some timing, see section 2.6

Sometimes it is useful to compute a Poincaré series only up to some degree.

Definition 1.122 *Let $R := k[x_1, \dots, x_n]$ be graded by the system of weights $\mathbf{w}$ of type r, $\tau \in \mathrm{TO}(\mathbb{Z}^r)$ and $M \subset R^r$ be an homogeneous R module. Let $\mathbf{d} \in \mathbb{Z}^r$. Then the* truncated Hilbert function *of M up to degree $\mathbf{d}$ is the function*

$$\mathcal{H}_M^{\mathbf{w},\mathbf{d}} : \mathbb{N}^r \longrightarrow \mathbb{N}$$

$$\mathbf{j} \rightsquigarrow \begin{cases} \mathcal{H}_M^{\mathbf{w}}(\mathbf{j}) & \text{for each } \mathbf{j} \in \mathbb{N}^r \text{ and } \mathbf{j} \leq_\tau \mathbf{d} \\ 0 & \text{otherwise} \end{cases}$$

and the truncated Poincaré series *of M up to degree $\mathbf{d}$ is*

$$\mathcal{HP}_M^{\mathbf{w},\mathbf{d}}(\lambda) := \sum_{\mathbf{d} \in \mathbb{N}^m} \mathcal{H}_M^{\mathbf{w},\mathbf{d}}(\mathbf{d})\lambda^{\mathbf{d}} \in \mathbb{Z}[[\lambda_1, \dots, \lambda_m]]$$

The truncated $\mathcal{HP}$-numerator of M with respect to $\mathbf{w}$ is $\langle M \rangle^{\mathbf{w},\mathbf{d}}$

Remark 1.123 *If the type of $\mathbf{w}$ is not small, then the computation of truncated Poincaré series may be substantially cheaper than the computation of the whole Poincaré series, see section 2.6. At least some tricks that may be applied are:*

1. *eliminate in the input all the terms which $\mathbf{w}$ degree is bigger than the truncation.*
2. *During the computation, eliminate all the branching multiplied by a $\mathbf{w}$ degree bigger than the truncation $\mathbf{d}$.*
3. *During the computation of $\langle I \rangle$, don't add terms of $\mathbf{w}$ degree lesser than $\mathbf{d}$.*

No comprehensive examination has been done on pivot strategies for the truncated algorithm.

Remark 1.124 *It may be of interest to compute the terms of $\langle \cdot \rangle$ only starting from some given $\mathbf{w}$ degree, or directly the value of the Hilbert function in $\mathbf{d}$ without overhead. It seems that, using some modification of the algorithm here discussed, the savings are not sufficient to justify the increased complexity of the algorithm, but no exhaustive examination has been performed.*

1.4.4 Dimension and multiplicity

An exposition of the theory of dimension may be found in [ZS] or [M]. We are here only interested in some brief remarks on the practical computation of dimension and multiplicity.

Proposition 1.125 *Let $R := k[x_1, \ldots, x_n]$, $\mathbf{w}$ be a system of weights, $\tau \in \mathrm{TO}(R)$ and I be a $\mathbf{w}$ homogeneous ideal of R. Let $J := \mathrm{SQFR}(\mathrm{Lt}_\tau(I))$,*

$$P^{\mathbf{w}}_{R/I}(\lambda) = \frac{\langle I\rangle^{\mathbf{w}}}{\prod_i (1-\lambda^{\partial_{\mathbf{w}}(x_i)})}(\lambda, 1, \ldots, 1) = \frac{Q(\lambda)}{(1-\lambda)^d} \text{ with } Q(1) \neq 0$$

$$P^{\mathbf{w}}_{R/J}(\lambda) = \frac{\langle J\rangle^{\mathbf{w}}}{\prod_i (1-\lambda^{\partial_{\mathbf{w}}(x_i)})}(\lambda, 1, \ldots, 1) = \frac{Q'(\lambda)}{(1-\lambda)^{d'}} \text{ with } Q'(1) \neq 0$$

Then, we have $\dim(R/I) = d = d'$ *and* $\mathrm{mult}(R/I) = Q'(1)$.

Proof. This is a classical result. For the multigraded case, see 2.31. □

The computation of dimension and multiplicity of an ideal is then immediate if the Poincaré series is known.

There is another algorithm for the computation of the dimension of an ideal that may be useful since it gives explicitly a maximal set of algebraically independent variables. It is usually slower than the one based on Hilbert function.

Lemma 1.126 *With the notation of the proposition above, we remark that a the complement with respect to $\{x_1, \ldots, x_n\}$ of minimal set of variables $\mathcal{S}$ with the property*

P: every element of a reduced list of generators of J contains a variable in $\mathcal{S}$

is a maximal set of algebraically independent variables modulo the ideal J, hence I.

Proof. See, for example, [BW], chapter 9. □

Remark 1.127 *It is easy to find a maximal set of variables satisfying the property P by a recursive back-tracking algorithm, see [BW], chapter 9.*

Remark 1.128 *In particular, $k[x_1, \ldots, x_n]/I$ is zero-dimensional if and only if for every variable there exists an element of the Gröbner basis beginning with a pure power of this variable. Detection of zero-dimensional ideals is then easy during the computation of any of theirs Gröbner basis.*

Remark 1.129 *The computational cost of the computation of the dimension is negligible with respect to the cost of the Gröbner basis computation which gives $Lt(I)$.*

Remark 1.130 *Everything said in this subsection holds for non-homogeneous ideals if the Gröbner basis is computed with respect to a degree-compatible term-ordering.*

Proof. We can homogenize the ideal first, and then compute dimension and multiplicity. These are invariant with respect to the homogeneization, see [ZS]. □

Remark 1.131 *For standard graded polynomial rings, it is well known that the Hilbert Function assumes definitely the same values as a polynomial function, the Hilbert Polynomial. If the graduation is non standard, it is easy to see that there exists finitely many polynomials that give the function values definitely. Moreover, there is a simple description of these polynomials using the unity roots. The details are a bit intricate, but quite elementary.*

Chapter 2

Gröbner Bases

In this chapter we briefly introduce Gröbner bases theory. The usual refinements of the Buchberger algorithm are noted, but not discussed. The variants of the Buchberger algorithm examined in detail are

- the homogeneous variant, section 2.2;
- the Buchberger algorithm for modules, section 2.3;
- the factoring Buchberger variant, section 2.4;
- the Hilbert-Driven and BiHilbert-Driven variants, section 2.5-2.6 and
- the Dynamic variant, section 2.7.

The homogeneous variant is discussed mainly to set the notation for the Hilbert variants and for the resolution algorithm discussed in chapter 3.

2.1 Gröbner bases and Buchberger algorithm

Let $R := k[x_1, \dots, x_n]$, I be an ideal of R and $\tau \in \mathrm{TO}(R)$. Among its system of generators, some are particularly useful for the effective computation of the algebraic operation on I.

Definition 2.1 *Let $R := k[x_1, \dots, x_n]$, $F := \{f_1, \dots, f_r\} \subset R$ and τ be a term-ordering on R. If $(\mathrm{Lt}_\tau(f_1), \dots, \mathrm{Lt}_\tau(f_r)) = \mathrm{Lt}_\tau(I)$, then $\{f_1, \dots, f_r\}$ is a* Gröbner basis *of the ideal (F) with respect to τ.*

There are several equivalent definition of Gröbner basis (see [Bu, MM, R3]). We are here interested in a definition which allows us an easy construction of Gröbner basis.

The notion of *normal form* is crucial for the computation of Gröbner basis.

Definition 2.2 *Let $f, g \in R := k[x_1, \dots, x_n]$ and $\tau \in \mathrm{TO}(R)$. Then the polynomial*

$$\mathrm{SPoly}\,(f,g) := \frac{Lc_\tau(g)\,\mathrm{lcm}\,(\mathrm{Lt}_\tau(f), \mathrm{Lt}_\tau(g))}{\mathrm{Lt}_\tau(f)} f - \frac{Lc_\tau(g)\,\mathrm{lcm}\,(\mathrm{Lt}_\tau(f), \mathrm{Lt}_\tau(g))}{\mathrm{Lt}_\tau(g)} g.$$

is the S-Polynomial *of the pair f, g.*

The procedure NF gives a reduced form of a polynomial with respect to a list of polynomials and a term-ordering.

Procedure NF

Input $f \in R$, $F := \{f_1, \dots, f_r\} \subset R$, $\tau \in \mathrm{TO}(R)$.
Output $g \in R$.

begin
  h:=0;
  begin
    while $\exists g \in F$ s.t. $\mathrm{Lt}_\tau(g) \mid \mathrm{Lt}_\tau(f)$ **do**
      begin
        choose $g \in F$ s.t. there exists $t \in T^n$ s.t. $\mathrm{Lt}_\tau(g) = t \cdot \mathrm{Lt}_\tau(f)$;
        $f := f - \frac{\mathrm{lc}(f)}{\mathrm{lc}(g)} t \cdot g$;
      endwhile;
    if $f \neq 0$ **then**
      begin
        $h := h + \mathrm{Lt}_\tau(f)$;
        $f := f - \mathrm{Lt}_\tau(f)$;
      endif;
  return h;
end.

The procedure NF stops for each f, F, τ, since the leading terms of f keep decreasing with respect to τ at each iteration of the main `while` of the procedure and τ is a term-ordering.

The above description of the procedure NF is not complete. In fact thre are in general several different choices for the polynomial g which divides f. For each possible choice (for a given input), we have a different procedure. The number of possible choices is finite. Let $S := \{A^1, \dots, A^v\}$ be the set of such procedures. Then, we have the function

$$\begin{array}{rccc} \mathrm{NF}_{F,\tau}: & R & \longrightarrow & \wp(R) \\ & f & \rightsquigarrow & \{A^1_{F,\tau}(f), \dots, A^v_{F,\tau}(f)\} \end{array}$$

Definition 2.3 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and $F := \{f_1, \dots, f_k\} \subset R$ The set $\mathrm{NF}_{F,\tau}(f) \subset R$ is the* set of the normal forms *of f with respect to $\{f_1, \dots, f_r\}$ and τ. If $\mathrm{NF}_{F,\tau}(f)$ is a singleton, the element $h \in \mathrm{NF}_{F,\tau}(f)$ is the* normal form *of f with respect to F, τ. $\mathrm{NF}_{F,\tau}(f)$. f is reduced to h through $\{f_1, \dots, f_r\}, \tau$, or $f @ > \{f_1, \dots, f_r\}, \tau >> h$. $\{f_1, \dots, f_r\}$ and τ are omitted if the notation in unambiguous.*

Theorem 2.4 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and $(F) := (f_1, \dots, f_r)$ be an ideal of R. Then the subsequent propositions are equivalent:*

- *$f_1, \dots, f_r$ is a Gröbner basis of (F) with respect to τ.*
- *For each $f \in R$, the set $\mathrm{NF}_{F,\tau}(f)$ is a singleton.*
- *For each pair $f, g \in \{f_1, \dots, f_r\}$, $\mathrm{NF}_{F,\tau}(\mathrm{SPoly}\,(f, g)) = 0$*

Definition 2.5 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$, $F := \{f_1, \dots, f_r\} \subset R$ and $I := (F)$. Then the list F is* interreduced *with respect to τ if, after the substitution of every polynomial f_i with $\mathrm{NF}_{F',\tau}(f)$, where $F' := F - \{f_i\}$ the list does not change.*

The procedure that transforms a polynomial list $F := f_1, \dots, f_r$ in a polynomial list $\overline{F}$ interreduced and such that $(F) = (\overline{F})$ is the *Interreduce* procedure. Its implementation is straightforward.

Since the normal form is in general not unique, the interreduced form of a polynomial list is in general not unique.

Definition 2.6 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and $F := \{f_1, \dots, f_r\}$. Let G be a Gröbner basis of (F) with respect to τ. If G is interreduced, and its elements are monic, then G is a* reduced Gröbner basis *of (F) with respect to τ.*

Theorem 2.7 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and $F := \{f_1, \dots, f_r\}$. Then the reduced Gröbner basis of (F) with respect to τ, $GB((F), \tau)$, is unique.*

Proof. See e. g. [Bu]. □

2.1.1 Buchberger algorithm

The Buchberger algorithm computes the Gröbner basis of a list of polynomials with respect to a term-ordering. Here it is a brief description of the Buchberger algorithm, and then its pseudo-code.
The input is the polynomial list $G := \{g_1, \dots, g_r\}$, of which we want to compute the Gröbner basis, and the term-ordering τ, with respect to which we want the basis. Then we create the pair list $L := \{(g_i, g_j) \mid i \leq j\}$. At this point, a pair $(g_i, g_j) \in L$ is chosen and deleted from the list. We then compute f:=NF $_{G,\tau}(\mathrm{SPoly}\,(g_i, g_j))$. If $f = 0$ another pair in L is chosen, and the cycle begins anew. If $f \neq 0$, the polynomial f is added to the list G, to the list L, are added the pairs (g_i, f), and the cycle begins anew. The cycle stops when the list L is empty. When the cycle stops, by proposition 2.4 in the list G there is a Gröbner basis of (G) with respect to τ. If a reduced Gröbner basis is wanted, the interreduction of G is then performed.is guaranteed by the noetherianity of R (or, if you prefer, by Dickson's lemma), since new elements not belonging to $\{\mathrm{Lt}_\tau(g) \mid f \in G\}$ are added to $\mathrm{Lt}_\tau(G)$ at every cycle that does not delete an element from L.

Some notation: let (g_i, g_j) a pair in L. Then $\mathrm{Lt}(i,j) := \mathrm{lcm}\,(\mathrm{Lt}(g_i), \mathrm{Lt}(g_j))$ and $\mathrm{Lt}(i,j,k) := \mathrm{lcm}\,(\mathrm{Lt}(g_i), \mathrm{Lt}(g_j), \mathrm{Lt}(g_k))$.

Buchberger algorithm

Input$G := \{f_1, \dots, f_r\} \subseteq R$, $\tau \in \mathrm{TO}(R)$;
Outputthe reduced Gröbner basis of (G) with respect to τ;

begin
  L:= generates the initial pairs of G;
  while $L \neq \emptyset$ **do begin**
    choose $(g_i, g_j) \in L$;
    L:=$L - \{(g_i, g_j)\}$;
    if $\mathrm{Lt}_\tau(g_i)$, $\mathrm{Lt}_\tau(g_j)$ are not coprimes **then**
      begin
        $g :=$ NF $_{G,\tau}(\mathrm{SPoly}\,(g_i, g_j))$;
        if $g \neq 0$ **then**
          begin
            update the pairs in L with respect to G and g;
            G:=$G \cup \{g\}$;
          endif;
      endif;
  endwhile;
  return *Interreduce*(G);
end.

Remark 2.8 *If $\mathrm{Lt}_\tau(g_i)$, $\mathrm{Lt}_\tau(g_j)$ are coprimes, then* NF $_{G,\tau}(\mathrm{SPoly}\,(g_i, g_j)) = 0$. *This is the* coprimality criterion *for the pairs.*

Remark 2.9 *During the execution of the Buchberger algorithm almost all the pairs are reduced to zero. It is then quite useful to have some criteria that predict the reduction to zero of a pair. The more efficient by far are the Gebauer-Möller criteria [GM]. Their use is necessary in practice to have terminating algorithms. Since all the variants here shown use these criteria, and don't change them except for some redefinition of the divisibility test, we note here only their use and don't give any detail. Other criteria to predict the reduction to zero of pairs which escaped the Gebauer-Möller criteria are shown in the Hilbert-Driven (section 2.5) and Bi-Hilbert-Driven (section 2.6) variants.*

The execution time of the Buchberger algorithm is strongly influenced by the term-ordering with respect to which we want to compute the basis and from the choice strategies for pairs and reductors. A variant of the Buchberger algorithm that exploits the sensibility to orderings changing, the Dynamic Algorithm, is shown in section 2.7

Proposition 2.10 *Let $R := k[x_1, \dots, x_n]$ and $I := (f_1, \dots, f_k)$ be an ideal of R. Then the degree of the polynomials in a Gröbner basis of I is bounded by*

$$((n+1)(d+1)+1)^{(n+1)2^{s+1}}$$

where s is the Krull dimension of I and d is the maximum degree of the polynomials f_i.

Proof. See [Metc] □

In practice, it is uncommon for a computation to reach the theoretical complexity, albeit there are examples that reach it, and several meaningful computations are possible. A prediction of the practical complexity of a given Gröbner basis computation is very difficult, as shown in the subsequent examples, but several good rules of thumb are known.

After much sperimentation, the implementers have reached an accord upon the subsequent facts, some very easy to observe, some more obscure :

- the coefficients and the support of the polynomials involved usually grow during the computations, even if in the final basis they are quite small.
- The pair selection strategy is crucial.
- The reductor selection strategy usually influences only the growth of the coefficient during the reduction computation.
- The choice of the term-ordering is crucial. Different term-orderings give very different Gröbner basis.
- Degree compatible term-orderings, and in particular the degrevlex term-ordering, usually give the fastest computations.
- Lexicographic term-orderings give usually very hard computations. The growth of the coefficient and of the supports during the computations is quite big, even if the final basis is quite simple.

Our work in centered on the optimizations of the ideal-module operations, and thus the details of the optimization of the Buchberger algorithm for the general case are not of interest here. We only note that

- The sugar pair selection strategy, in one of its flavor, is usually useful [GMNRT].

For an analysis of the strategies, see [GMNRT].

Hence, in this work, the Buchberger algorithm is supposed to use the Gebauer-Möller criteria and the sugar pair selection strategy. The reductors are chosen by age : the reductor with the lower index in the basis is used.

2.2 Homogeneous Gröbner bases

The Buchberger algorithm can exploit the homogeneity of an ideal in several ways. There is no major difference in the performance of the algorithm, but the data structure has to be arranged to exploit fully the homogeneity. Moreover, homogeneous ideals are usually easier to compute and less sensitive to the order in which the pairs are processed.

Remark 2.11 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights and I be an ideal of R. Then I is $\mathbf{w}$ homogeneous if the list given by the interreduction of the generators of I is composed only by $\mathbf{w}$ homogeneous polynomials.*

2.2.1 The homogeneous Buchberger algorithm

Proposition 2.12 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ a system of weights of type r, $\tau \in \mathrm{TO}(R)$, $\sigma \in \mathrm{TO}(\mathbb{N}^r)$ and $f, g, f_1, \dots, f_r$ $\mathbf{w}$ homogeneous polynomials of R. Then*

a) $\partial_{\mathbf{w}}(\mathrm{SPoly}\,(f,g)) >_\sigma Max_\sigma(\partial_{\mathbf{w}}(f), \partial_{\mathbf{w}}(g))$.

b) $f @ > \{f_1, \dots, f_r\}, \tau >> h \Longrightarrow h$ *is* $\mathbf{w}$ *homogeneous.*

c) S-Poly(f,g) is $\mathbf{w}$ homogeneous.

d) f can be reduced only by $\mathbf{w}$ homogeneous polynomials of uniformly lesser $\mathbf{w}$ degree.

Proof. Easy check. □

If the input polynomials are $\mathbf{w}$ homogeneous, the cases *b)* and *c)* guarantee that every polynomial processed in the Buchberger algorithm is $\mathbf{w}$ homogeneous. Case *a)* suggests that the polynomial should be kept in increasing $\mathbf{w}$ order. Case *d)* allows us us to reduce substantially the number of candidate divisors for a polynomial, giving a faster reductor search. The procedure $\mathrm{NF}^{\mathbf{w}}$ is a version of NF that search for possible reductors only among polynomials whose $\mathbf{w}$ degree is uniformly lesser than the $\mathbf{w}$ degree of the polynomial being reduced. From what said above, it is obvious that in the homogeneous Buchberger algorithm, the pairs should be processed and kept in increasing $\mathbf{w}$ degree order, and the elements of F should be kept in increasing $\mathbf{w}$ degree order.

Remark 2.13 *To optimize the search for reductors, and some other operations that are introduced in the Hilbert-Driven algorithm 2.5, the list is no more the most efficient structure for the pairs and the reductors.*

Definition 2.14 *Let $R := k[x_1, \dots, x_n]$ and $\mathbf{w}$ be a system of weights of type r. Then a $\mathbf{w}$-list is*

- *an $\mathbf{w}$-list of level 0 is a list of $\mathbf{w}$ homogeneous elements ordered by increasing w_r degrees.*
- *a $\mathbf{w}$-list of level i, $i : 1 \dots r$ is a list of $\mathbf{w}$ lists of level $i-1$ ordered by increasing w_i degrees.*

Example 2.15 *Let $R := k[x,y,z]$ and $\mathbf{w} := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ a system of weights for R. Let $f_1 := x^4, f_2 := x^3y, f_3 := x^2y^2, f_4 := x^7, f_5 := y^7$. Then the $\mathbf{w}$ list of level 2 of $f_1, \dots, f_5$ is $[[f_1, f_2, f_3], [f_4], [f_5]]$.*

If reductors are kept in $\mathbf{w}$ lists, then the search for reductors is much easier. It is not clear if this fast reductor search counterbalances the increased complexity of the data structure. In cases where the number of reductor is very high (e. g. binomial ideals), the savings may be relevant. Further experiments are needed.

Homogeneous Buchberger Algorithm

Let $\mathbf{d}$ be the least $\mathbf{w}$ degree of the pairs in L. When the pairs in L degree $\mathbf{d}$ are reduced with respect to F, τ, the results are kept in an auxiliary list F'. When all the pairs of degree $\mathbf{d}$ have been processed, the polynomials in F' are interreduced, the pairs of the resulting polynomials are added to L and the polynomial themselves are added to F. The list F' is emptied. This approach allows us us to use linear algebra for the interreduction of the polynomials of degree $\mathbf{d}$, and avoid the final interreduction of F. Moreover, by *d)* of proposition 2.12, the only polynomials that may divide the polynomials of $\mathbf{w}$ degree $\mathbf{d}$ are polynomials whose $\mathbf{w}$ degree is uniformly lesser than $\mathbf{d}$.

Input $G := \{g_1, \dots, g_r\} \subseteq k[\mathbf{x}]$, homogeneous, $\tau \in \mathrm{TO}(R)$ and $\mathbf{w}$, system of weights for R.
Output the reduced Gröbner basis of G with respect to τ;.

begin
  L:= generates the initial pairs;
  d:=$min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$;
  while $L \neq \emptyset$ **do begin**
    choose $(g_i, g_j) \in L$ s.t. $d = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
    $L := L - \{(g_i, g_j)\}$;
    $g := \mathrm{NF}_{\sigma}(\mathrm{SPoly}\,(g_i, g_j), G)$;
    if $g \neq 0$ **then** G':=$G' \cup \{g\}$;
    if $d \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
      begin
        G':=*Interreduce*(G');
        L,G:=`Upgrade`(L, G', G);
        L':=$\emptyset$;
        d:=$min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$;
      endif;
  endwhile;
  return G;
end.

Definition 2.16 *Let* $R := k[x_1, \dots, x_n]$ *be a* $\mathbf{w}$ *graded ring,* $\tau \in \mathrm{TO}(R)$*,* I *be a* $\mathbf{w}$*-homogeneous ideal,* $G := \{g_1, \dots, g_k\}$ *a* $\mathbf{w}$*-homogeneous system of generators of* I *and* $\mathbf{d} \in \mathbb{N}^m$*. Then* G *is a* $\mathbf{d}$-truncated *Gröbner basis of* I *if* $\{\mathrm{Lt}_\tau(g)s.\ t.f \in F_{\leq_\tau \mathbf{d}}\}$ *generates* $\mathrm{Lt}_\tau\,(I)_{\leq_\tau \mathbf{d}}$.

Remark 2.17 *To compute a Gröbner basis of* I *truncated to some* $\mathbf{w}$ *degree* $\mathbf{d}$*, it suffices to introduce a check in the homogeneous Buchberger algorithm that stops the computation after the current pairs reach a degree bigger than* $\mathbf{d}$*. Moreover, no pairs of* $\mathbf{w}$ *degree bigger than* $\mathbf{d}$ *are created.*

2.3 Module Gröbner bases

Module Gröbner bases are defined in exactly the same way as ring Gröbner bases.

Definition 2.18 *Let* $R := k[x_1, \dots, x_n]$ *and* $M \subset R^r$ *an* R *module and* $\mu \in \mathrm{MTO}(M)$*. Then* $F := \{m_1, \dots, m_k\}$ *is a Gröbner basis for* M *if* $\mathrm{Lt}_\mu(M) = \{\mathrm{Lt}_\mu(m_1), \dots, \mathrm{Lt}_\mu(m_k)\}$

Everything shown in section 2.1 has a module counterpart, with the only difference being that if two module elements have coprime module terms, their SPolynomial not necessarily reduces to 0.

Example 2.19 *The vector list* $([x_1^2, 1], [x_2, 0]) \subset (k[x_1, x_2])^2$ *is not a Gröbner basis of the submodule that it generates with respect to the* $\mathrm{lex}(x_1 > x_2) + Pos^+$ *module term-ordering. The addition of* $[x_2, 0]$ *is necessary.*

It is hence possible to have a module Buchberger algorithm that mimicks exactly the ring Buchberger algorithm. But this leads to the duplication of the implementation.

Another approach is to implement the module Buchberger algorithm only and to consider ideals as special modules, like the macaulay system.

The last approach is to implement the ring Buchberger algorithm only and to consider modules as special ideals. We can use use the embedding introduced in subsection 1.3.1, and work not in M but in $\phi(M)$ in a suitable overring. We want to show that using this approach, no modification is necessary to the Buchberger algorithm for the computation of the Gröbner basis of a module, if the terms divisibility is interpreted as in definition 1.83, and the coprimality criterion is not used.

Let $R := k[x_1, \dots, x_n]$, $\mu \in \mathrm{TO}(M)$ and let $M \subset R^r$ be a R module generated by $\{n_1, \dots, n_p\}$. Let $E_1, \dots, E_r$ be new variables and $t_1, \dots, t_r$ terms of $k[\mathbf{x}, \mathbf{E}]$ linearly independents over $k[\mathbf{x}]$. Let module morphism

$$\begin{array}{cccc} \phi: & R^r & \longrightarrow & k[\mathbf{x}, \mathbf{E}] \\ & \mathbf{e}_i & \rightsquigarrow & t_i \end{array}$$

fulfilling the conditions of theorem 1.88. Then employing the term division definition introduced in 1.83, and considering GCD and lcm in the sense of modules, it follows that :

1) given elements $a, b \in M$, a divides b iff exists $c \in R$, $ca = b$;
2) a common multiple of a and b may not exist; hence the S-polynomial of two elements of M may not exist, and the corresponding critical pairs can be discarded;
3) the GCD of a and b in M not only may not exist, but it is never 1, the multiplicative identity, that does not exist in M. The corresponding criterion that allows us us to discard pairs in the ring case is never verified in the module case.

Then,

1) a critical pair $(a\, \mathbf{x}^I \mathbf{E}^J + \cdots, b\, \mathbf{x}^H \mathbf{E}^K + \cdots)$ is considered only if $J = K$.
2) When simplifying a polynomial $a\, \mathbf{x}^I \mathbf{E}^J + \cdots$, we consider only simplifiers g_j such that $\mathrm{Lt}(g_j) = \mathbf{x}^L \mathbf{E}^J$.
3) discarding useless pairs, we do not discard (g_i, g_j) if $\mathrm{Lt}(g_i)$ and $\mathrm{Lt}(g_j)$ are coprime.

Proposition 2.20 *With the notation introduced above, the ring Buchberger algorithm working in $k[\mathbf{x}, \mathbf{E}]$ computes the Gröbner basis of $\phi(M)$ as a module with the operations modified as explained in 1-3 above.*

Proof. Follows from the discussion above. □

Example 2.21 *Let $M \subset (k[x_1, x_2])^2$ be a $k[x_1, x_2]$-module generated by the elements $(< x_1^2, 1 >, < x_1x_2 + x_2^2, 0 >, < 0, x_2 >)$. The module term-ordering on M is the $\mathrm{lex}(x_1 > x_2) + $ Position module term-ordering, e. g. the module term-ordering μ defined by*

$$x_1^\alpha x_2^\beta \mathbf{e}_i >_\mu x_1^{\alpha'} x_2^{\beta'} \mathbf{e}_j \Longleftrightarrow x_1^\alpha x_2^\beta >_{lex} x_1^{\alpha'} x_2^{\beta'} \text{ or } x_1^\alpha x_2^\beta = x_1^{\alpha'} x_2^{\beta'} \text{ and } i > j'$$

The embedding is

$$\begin{array}{cccc} \phi: & (k[x_1, x_2])^2 & \longrightarrow & k[x_1, x_2, E] \\ & \mathbf{e}_1 & \rightsquigarrow & 1 \\ & \mathbf{e}_2 & \rightsquigarrow & E \end{array}$$

and on $k[x_1, x_2, E]$ there is the term-ordering, compatible with ϕ, μ defined by the matrix

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

The image of the generators of M is

$$\phi(M) = \{f_1 := x_1^2 + E, f_2 := x_1x_2 + x_2^2, f_3 := x_2E\}$$

The pairs (f_1, f_3) and (f_2, f_3) are not considered, since the exponent of E in their leading terms is different. The pair (f_1, f_2) is considered, giving.

$$SPoly(f_1, f_2) = -x_1x_2^2 + x_2E =: g$$

Note that $\mathrm{Lt}(f_2) \mid \mathrm{Lt}(g)$, *while* $\mathrm{Lt}(f_3) = x_2E \nmid x_1x_2^2 = \mathrm{Lt}(g)$, *since the exponent E in their leading terms is different.*

$$g + x_2f_2 = x_2^3 + x_2E - f_3 = x_2^3$$

and no further reduction is possible. The polynomial $f_4 := x_2^3$ is an element of the Gröbner basis of M. The pair (f_3, f_4) is not considered, while the pairs (f_1, f_4) and (f_1, f_4) are, giving

$$SPoly(f_1, f_4) = x_2^3f_1 - x_1^2f_4 = x_2^3E@ > f_3 >> 0;$$

$$SPoly(f_1, f_4) = x_2^2f_2 - x_1f_4 = x_2^4@ > f_4 >> 0.$$

Thus the Buchberger algorithm for modules, applied to the generators

$$\{x_1^2 + E, x_1x_2 + x_2^2, x_2E\}$$

gives

$$L := \{x_1^2 + E, x_1x_2 + x_2^2, x_2E, x_2^3\}$$

hence the list

$$\phi^{-1}(L) = \{< x_1^2, 1 >, < x_1x_2 + x_2^2, 0 >, < 0, x_2 >, < x_2^3, 0 >\}$$

is the reduced Gröbner basis of M with respect to μ.

Example 2.22 *Let R be $k[x, y, z]$ and $\mu := \mathrm{degrevlex}(x > y > z) \in \mathrm{TO}(R)$. Let τ be the module term-ordering on R^2 defined by*

$$m\mathbf{e}_i >_\mu m'\mathbf{e}_j \iff m >_\tau m' \text{ or } m = m' \text{ and } i > j.$$

Let the module morphism ϕ be defined by

$$\begin{array}{ccc} R^2 & @ > \phi >> & k[\mathbf{x}, E] \\ \mathbf{e}_1 & \rightsquigarrow & E \\ \mathbf{e}_2 & \rightsquigarrow & E^2, \end{array}$$

and the term-ordering μ' on $k[\mathbf{x}, E]$ defined by the matrix

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is immediate to check that ϕ is an injective and ordered embedding of R^2 in $k[x, y, z, E]$. Then the generators of the submodule Let $M := ([x^2 - y^2, xy - yz], [x^3 - z^3, xy - z^2], [0, xy + z]) \subseteq R^2$ be a submodule of R^2. Then

$$\phi(M) = (x^2E + xyE^2 - y^2E - yzE^2, x^3E - z^3E + xyE^2 - z^2E^2, xyE^2 + zE^2)$$

The Gröbner basis of $\phi(M)$ in $k[x,y,z,E]$ with respect to μ', computed with the Buchberger algorithm modified as above, is

$$
\begin{aligned}
F := \{ \quad & x^2E - y^2E - yzE^2 - zE^2, \\
& xyE^2 + zE^2, \\
& xy^2E - z^3E + xzE^2 - 2z^2E^2 - zE^2, \\
& y^4E + y^3zE^2 - xz^3E + x^2zE^2 + y^2zE^2 - 2xz^2E^2 - xzE^2, \\
& yz^4E^2 - x^3zE^2 + 2x^2z^2E^2 - y^2z^2E^2 + z^4E^2 + x^2zE^2 - y^2zE^2, \\
& x^4zE^2 - 2x^3z^2E^2 - xz^4E^2 + z^5E^2 - x^3zE^2 - yz^3E^2 - yz^2E^2 \ \},
\end{aligned}
$$

The counterimage of $\phi(M)$

$$
\begin{aligned}
\phi^{-1}(F) = \{ \quad & [x^2 - y^2, -yz - z], \\
& [0, xy + z], \\
& [xy^2 - z^3, xz - 2z^2 - z], \\
& [y^4 - xz^3, y^3z + x^2z + y^2z - 2xz^2 - xz], \\
& [0, yz^4 - x^3z + 2x^2z^2 - y^2z^2 + z^4 + x^2z - y^2z], \\
& [0, x^4z - 2x^3z^2 - xz^4 + z^5 - x^3z - yz^3 - yz^2] \ \}
\end{aligned}
$$

is the Gröbner basis of M with respect to μ.

2.4 Factorizing Gröbner

In several instances, we are not interested in the computation of the Gröbner basis of an ideal I, but in the computation a set of Gröbner bases of ideals such that the union of their associated varieties is the variety of I; or equivalently, of ideals such that their intersection has the same radical as the original basis.

In these cases, if during the computation it happens to recognize that a polynomial g factorize as $g_1 \cdots g_k$, it is useful to split the computation in k pieces, using the *Factorizing Buchberger Algorithm*

The factorizing Buchberger algorithm consists in maintaining a list of ideals, and computing the Gröbner basis of each of them with the pattern: whenever we compute a new polynomial belonging to the ideal and we find that is reducible, consider its distinct factors and split the ideal into as many ideals as there are different factors, adding to each one of them one of the factors. And insert these ideals in the list in place of the original ideal.

There are at least three main possible interpretations of "finding" that a polynomial is reducible:

- actively try to factor any polynomial (maybe only at selected points of the algorithm) with a general-purpose factorizer that can exhibit any existing factorization;

- rely on GCD for discovering factorizations: maintain a list of test polynomials (they may be e.g. the variables, or the list of basis elements, etc.) and try the GCD of them with any new polynomial; if the GCD is non-trivial, try to factor as much as possible the polynomial;

- try a factorization only if you stump into it: whenever something happens that cannot happen if a polynomial is irreducible, or an ideal is prime, (like e.g. having to invert an element that happens to be a zero-divisor) patch the situation with a splitting.

We will usually adopt this latter attitude in the description of our algorithms; however the other approaches are often viable too. The problem is usually that a lot of time is spent in trying to factor irreducible polynomials, but whenever a factorization is found the savings are very large. For this type of factorizations a heuristic irreducibility test is very useful. See [D] for an account.

2.5 Hilbert-Driven Buchberger Algorithm

If we want to compute the Gröbner basis of an homogeneous ideal, and we knew in advance the Hilbert function, then substantial savings are possible exploiting the information that the Hilbert function gives on the structure of the ideal during the execution of Buchberger algorithm.

Proposition 2.23 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w} := (w_1, \dots, w_m)$ be a system of weights, I,J two $\mathbf{w}$-homogeneous ideals of R such that $J \subseteq I$ and $\tau \in \mathrm{TO}(T^n)$. Let*

$$\langle I \rangle^{\mathbf{w}} = \sum_{u \in \mathbb{N}^m} a_u \lambda^u \text{ and } \langle J \rangle^{\mathbf{w}} = \sum_{u \in \mathbb{N}^m} b_u \lambda^u$$

then for each $\mathbf{d} \in \mathbb{N}^m$ the following conditions are equivalent:

i) $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r}) = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r})$ *for each* $\mathbf{r} <_\tau \mathbf{d}$ *and* $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{d}) < \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{d})$

ii) $a_{\boldsymbol{r}} = b_{\boldsymbol{r}}$ *for each* $\mathbf{r} <_\tau \mathbf{d}$ *and* $a_{\boldsymbol{d}} < b_{\boldsymbol{d}}$

Moreover, $dim_k(I_{\boldsymbol{d}}) - dim_k(J_{\boldsymbol{d}}) = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{d}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{d}) = b_{\boldsymbol{d}} - a_{\boldsymbol{d}}$.

Proof. We observe that

$$\sum_{\boldsymbol{r} \in \mathbb{N}^m} (b_{\boldsymbol{r}} - a_{\boldsymbol{r}}) \lambda^{\mathbf{r}} = (1 + f(\lambda)) \sum_{\boldsymbol{r} \in \mathbb{N}^m} (\mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r})) \lambda^{\mathbf{r}}$$

where $1 + f(\lambda) = \prod_{i=1}^n (1 - \lambda^{\partial_{\mathbf{w}}(x_i)}) \in \mathbb{Z}[\lambda]$ and $\partial_{\mathbf{w}}(t) > 0$ for each $t \in \mathrm{Supp}\,(F)$ (see Lemma 1.103). Then the conclusion follows easily $\square$

Definition 2.24 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w} := (w_1, \dots, w_m)$ be a system of weights and $\sigma \in \mathrm{TO}(\mathbb{Z}^m)$ Let F be a $\mathbf{w}$-homogeneous system of generators of I and*

$$\langle I \rangle^{\mathbf{w}} = \sum_{u \in \mathbb{N}^m} a_u \lambda^u \ , \ \langle \mathrm{Lt}_\tau(F) \rangle^{\mathbf{w}} = \sum_{u \in \mathbb{N}^m} b_u \lambda^u.$$

Then $Min(F, \mathbf{w}) := \min_\sigma \{\mathbf{d} \in \mathbb{N}^m$ s. t. $a_{\mathbf{d}} \neq b_{\mathbf{d}}\}$ is the (first) difference *of F with respect to* (F) *and* $\mathbf{w}$, *and the integer $b_{Min(F,\mathbf{w})} - a_{Min(F,\mathbf{w})}$ is the* (first) discrepancy.

Proposition 2.25 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights of type m, $\sigma \in \mathrm{TO}(\mathbb{Z}^m)$, I be a $\mathbf{w}$-homogeneous ideal, F a $\mathbf{w}$-homogeneous system of generators of I and τ a $\mathbf{w}$-compatible term-ordering; then:*

i) $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{d}) \leq \mathcal{H}^{\mathbf{w}}_{R/\mathrm{Lt}_\tau(F)}(\mathbf{d})$ *for every* $\mathbf{d} \in \mathbb{N}^m$

ii) $\mathcal{H}^{\mathbf{w}}_{R/I} = \mathcal{H}^{\mathbf{w}}_{R/\mathrm{Lt}_\tau(F)} \iff$ *F is a Gröbnerbasis of I*

iii) F is an $\mathbf{r}$-truncated Gröbner basis of I for every $\mathbf{r} <_\sigma Min(F, \mathbf{u})$

Exploiting proposition 2.25, if we know the Hilbert function of the ideal of which we want to compute the Gröbner basis, we can use a variant of the homogeneous Buchberger algorithm which uses such Hilbert function to predict the reduction to zero of several pairs.

The Hilbert-Driven Buchberger algorithm

Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights and $G := \{f_1, \dots, f_k\}$ a set of $\mathbf{w}$ homogeneous polynomials of R. Then, if we know $\langle (G) \rangle^{\mathbf{w}}$, At start, the first difference of G with respect to (G) and $\mathbf{w}$ is computed, let it be $\mathbf{d}$, and its discrepancy be m. Then the pairs of $\mathbf{w}$ degree $\mathbf{d}$ are processed until m pairs don't reduce to zero. At this point, all the remaining pairs in $\mathbf{w}$ degree $\mathbf{d}$ are forced to reduce to zero, and than can be deleted. The next difference and discrepancy are computed, and the algorithm goes on. When the first difference is zero, then G is a Gröbner basis of (G) with respect to τ by the proposition 2.25. All the remaining pairs are deleted, the optional interreduction occurs, and the algorithm stops.

Input

$G := \{g_1, \dots, g_r\} \subseteq k[\mathbf{x}]$, with g_i $\mathbf{w}$ homogeneous
$\tau \in \mathrm{TO}(R)$
$\mathbf{w}$, system of weights for R
$\langle (G) \rangle^{\mathbf{w}}$

Output

the reduced Gröbner basis of G with respect to τ.

begin
$L :=$ generates the initial pairs form G;
$\mathbf{d} := \partial_{\mathbf{w}}(\mathrm{in}(\langle \mathrm{Lt}_\tau(G) \rangle^{\mathbf{w}} - \langle (G) \rangle^{\mathbf{w}}))$;
Kill the pairs in L whose $\mathbf{w}$ degree is lesser than $\mathbf{d}$;
while $L \neq \emptyset$ **do begin**
choose $(g_i, g_j) \in L$ s.t. $\mathbf{d} = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
$L{:=}L - \{(g_i, g_j)\}$;
$f := \mathrm{NF}\ _\tau(\mathrm{SPoly}\,(g_i, g_j), F)$;
if $f \neq 0$ **then** $F'{:=}F' \cup \{h\}$;
if $\mathbf{d} \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
begin
$G'{:=}Interreduce(G')$;
L,G:=`Upgrade`$^{\mathbf{w},\mathbf{d}}(L, G', G)$;
$L'{:=}\emptyset$;
$\mathbf{d} := \mathrm{in}(\langle \mathrm{Lt}_\tau(G) \rangle^{\mathbf{w}} - \langle (G) \rangle^{\mathbf{w}})$;
endif;
Kill the pairs in L whose $\mathbf{w}$ degree is $\mathbf{w}$ lesser than $\mathbf{d}$;
endwhile;
return G;
end.

Remark 2.26 *The procedure* `Upgrade`$^{\mathbf{w},\mathbf{d}}$ *does not create pairs whose* $\mathbf{w}$ *degree is lesser than* $\mathbf{d}$*.*

The Hilbert-Driven Buchberger algorithm is useful whenever the Hilbert function of the input ideal is known. This is the case, for example, if the input polynomials are a regular sequence. In several other cases, usually coming from geometric constructions, it is possible to compute the Poincarè series of an ideal without the explicit determination of a system of generators. These results expand significatively the scope of application of Hilbert-Driven techniques. See [BCNR, RV].

Another very useful application of the Hilbert-Driven Buchberger Algorithm is the computation of the Gröbner basis of an ideal with respect to a term-ordering τ when the Gröbner basis of I with respect to a different term-ordering is already known. That change of basis application is quite useful for the radical algorithm discussed in chapter 5. Other methods are the FGLM algorithm for zero-dimensional ideals and the Gröbner walk techniques. The relative efficiency of these methods is still unclear. See [T], [AGK] and [FGLM].

Some example of the Hilbert-Driven algorithm are shown in subsection 2.6.2, together with Bi-Hilbert-Driven examples.

2.6 Bi-Hilbert-Driven Buchberger Algorithm

In general, using the Hilbert-Driven algorithm, a high type system of weights allows us us to detect more useless pairs that a low type system of weights. But for a high type system of weights the number of Poincaré series computation and the cost of each computation, is quite high.

Example 2.27 *Let* $R := \mathbb{Z}_{32003}[t, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, u]$, $\tau := \mathrm{lex}(t > \cdots > u)$ *and*

$$\begin{aligned} I := (&-tc^2e + 2tbcf - taf^2 - tb^2h + taeh + k^4, \\ &- tcde + tbdf + tbcg - tafg - tb^2i + taei + l^4, \\ &- td^2e + 2tbdg - tag^2 - tb^2j + taej + m^4, \\ &- tcdf + tc^2g + tbdh - tagh - tbci + tafi + n^4, \\ &- td^2f + tcdg + tbdi - tagi - tbcj + tafj + o^4, \\ &- tdf^2 + tcfg + tdeh - tbgh - tcei + tbfi + p^4, \\ &- tdfg + tcg^2 + tdei - tbgi - tcej + tbfj + q^4, \\ &- td^2h + 2tcdi - tai^2 - tc^2j + tahj + r^4, \\ &- tdgh + tdfi + tcgi - tbi^2 - tcfj + tbhj + s^4, \\ &- tg^2h + 2tfgi - tei^2 - tf^2j + tehj + u^4) \end{aligned}$$

The Gröbner basis of I *with respect to* τ *has* 421 *elements, the set of whose supports has* 15686 *terms. The computation take* 27 *s. The computation of the Poincaré series of* $\mathrm{Lt}_\tau(I)$

w.r.t the standard graduation takes 0.31 *s, and the univariate polynomial* $\langle I\rangle$, *of degree* 30, *has* 31 *terms.*

w.r.t. $\mathbf{w} := \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 4&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$ *takes* 0.36 *s, and the bivariate polynomial* $\langle I\rangle$, *of degree* 80, *has* 11 *terms.*

with respect to $\mathbf{w} := [1, 2, 1, 3, 1, 4, 1, 5, 1, 6, 1, 7, 1, 8, 1, 9, 1, 10, 1, 11, 1]$ *takes* 0.68 *s, and the univariate polynomial* $\langle I\rangle$, *of degree* 300, *has* 292 *terms.*

w.r.t. $\mathbf{w} := \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&2&3&40&5&6&7&8&9&1&2&30&4&5&6&70&8&90&0&0&0 \end{bmatrix}$ *takes* 15 *s, and the bivariate polynomial* $\langle I\rangle$, *of degree* 1568, *has* 32456 *terms.*

w.r.t. $\mathbf{w} := \begin{bmatrix} 1&1&1&3&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&2&3&40&5&6&7&8&9&1&2&30&4&5&6&70&8&90&0&0&0 \\ 1&7&3&40&5&4&7&8&0&1&2&30&0&5&6&70&8&90&0&0&0 \\ 1&7&3&40&51&4&7&8&0&11&2&30&0&5&6&70&100&90&0&0&0 \end{bmatrix}$
takes 1266 *s, and the pentavariate polynomial* $\langle I\rangle$, *of degree* 4640, *has* 646214 *terms.*
Remark that computation for high type systems of weights may be much more expensive than Gröbner basis computations even if the type doesn't approach the number of variables.

For more examples, where the systems of weights are more canonical, see subsection 2.6.2.

Remark 2.28 *The use of Kronecker's trick to keep a system of weights of type one doesn't solve the problem. While it is quite easy to apply Kronecker's trick, the integers in the system of weights became very large, and the same kind of patologies appears as in the multigraded case.*

An idea for avoid in part these problems, while reaping the benefits of high type systems of weights is to use a system of weights of type one, and then refine it to higher type system of weights.

Definition 2.29 *Let* $\mathbf{w} = (w_1, \dots, w_m)$ *be a system of weights and* $1 \le k < m$*. We say that* $\mathbf{u} := (w_1, \dots, w_k)$ *is a* subsystem of weights *of* $\mathbf{w}$ *if it is a system of weights. Under these circumstances we can also say that* $\mathbf{u} := (w_1, \dots, w_k)$ *is a system of weights and* $\mathbf{w} = (w_1, \dots, w_m)$ *is a* refinement *of* $\mathbf{u}$*. An term-ordering* μ compatible *with* $\mathbf{u}, \mathbf{w}$*, or simply with* $\mathbf{w}$ *(as subsystem of weights of* $\mathbf{u}$*) is such that there exist* $\tau \in \mathrm{TO}(\mathbb{Z}^k)$ *and* $\sigma \in \mathrm{TO}(\mathbb{Z}^{m-k})$ *s.t.* $\mu = \tau \times \sigma$*.*

Lemma 2.30 *Let* $\mathbf{u} = (w_1, \dots, w_k)$ *be a system of weights,* $\mathbf{w} := (w_1, \dots, w_m)$ *be a refinement of* $\mathbf{u}$ *and* $r \in \mathbb{N}^k$*. Then*

i) T_r^n*, the set of terms of* $\mathbf{u}$*-degree* r*, is the disjoint union of the* $T_{(r,s)}^n$*'s, the set of terms of* $\mathbf{w}$*-degree* (r,s)*, where* $s \in \mathbb{N}^{m-k}$

ii) Let I *be a* $\mathbf{w}$*-homogeneous ideal of* R*. Then* $(R/I)_r = \bigoplus_{s\in\mathbb{N}} (R/I)_{r,s}$*. Moreover, if* $\boldsymbol{\mathcal{H}}^{\mathbf{u}}_{R/I}$ *and* $\boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/I}$ *are the corresponding Hilbert functions, then*

$$\boldsymbol{\mathcal{H}}^{\mathbf{u}}_{R/I}(r) = \sum_{s\in\mathbb{N}^{m-k}} \boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/I}(r,s)$$

Proof. The first part is quite straightforward. As to the second one, we observe that a $\mathbf{w}$-homogeneous ideal of R is also $\mathbf{u}$-homogeneous. But $\mathbf{u}$ is a system of weights, hence each T_r^n is finite; this implies that, for a fixed $r \in \mathbb{N}^k$, $T_{(r,s)}^n$ is empty for almost all $s \in \mathbb{N}^{m-k}$. The conclusion follows. □

Lemma 2.31 *Let* $\mathbf{u} = (w_1, \dots, w_k)$ *be a system of weights,* $\mathbf{w} := (w_1, \dots, w_m)$ *be a refinement of* $\mathbf{u}$ *and* I *a* $\mathbf{w}$*-homogeneous ideal of* R*. Let*

$$\langle I \rangle^{\mathbf{w}} = \sum_{u\in\mathbb{N}^m} a_u \lambda^u, \langle I \rangle^{\mathbf{u}} = \sum_{v\in\mathbb{N}^k} a'_v \lambda^v$$

be the $\boldsymbol{\mathcal{HP}}$*-numerators of* I *with respect to* $\mathbf{w}$ *and* $\mathbf{u}$ *respectively. Then*

i) $\boldsymbol{\mathcal{HP}}^{\mathbf{u}}_{R/I}(\lambda_1, \dots, \lambda_k) = \boldsymbol{\mathcal{HP}}^{\mathbf{w}}_{R/I}(\lambda_1, \dots, \lambda_k, 1, \dots, 1)$.

ii) $\langle I \rangle^{\mathbf{u}} (\lambda_1, \dots, \lambda_k) = \langle I \rangle^{\mathbf{w}} (\lambda_1, \dots, \lambda_k, 1, \dots, 1)$

iii) $a'_v = \sum_{s\in\mathbb{N}^{m-k}} a_{(v,s)}$ *for every* $v \in \mathbb{N}^k$*.*

Proof. It is an easy consequence of the previous Lemma. □

Proposition 2.32 *Let* $\mathbf{u} = (w_1, \dots, w_k)$ *be a system of weights,* $\mathbf{w} := (w_1, \dots, w_m)$ *be a refinement of* $\mathbf{u}$ *and* $\mu \in \mathrm{TO}(\mathbb{Z}^m)$ *compatible with* $\mathbf{u}, \mathbf{w}$*. Let* $J \subset I$ *be two* $\mathbf{w}$*-homogeneous ideals of* R *such that*

$$\langle I \rangle^{\mathbf{w}} = \sum_{\boldsymbol{u}\in\mathbb{N}^m} a_{\boldsymbol{u}} \lambda^{\boldsymbol{u}} \ , \ \langle J \rangle^{\mathbf{w}} = \sum_{\boldsymbol{u}\in\mathbb{N}^m} b_{\boldsymbol{u}} \lambda^{\boldsymbol{u}}$$

and

$$\langle I \rangle^{\mathbf{u}} = \sum_{\boldsymbol{v}\in\mathbb{N}^k} a'_{\boldsymbol{v}} \lambda^{\boldsymbol{v}} \ , \ \langle J \rangle^{\mathbf{u}} = \sum_{\boldsymbol{v}\in\mathbb{N}^k} b'_{\boldsymbol{v}} \lambda^{\boldsymbol{v}}.$$

If $\boldsymbol{r} \in \mathbb{N}^k$ *is s.t.* $a'_{\boldsymbol{v}} = b'_{\boldsymbol{v}}$ *for* $\mathbf{v} <_\mu \mathbf{r}$ *and* $a'_{\boldsymbol{r}} < b'_{\boldsymbol{r}}$ *then*

i) $a_{(\boldsymbol{v},\boldsymbol{s})} = b_{(\boldsymbol{v},\boldsymbol{s})}$ *for each* $\mathbf{s} \in \mathbb{N}^{m-k}$ *and* $\mathbf{v} <_\mu \mathbf{r}$

ii) $b_{(\boldsymbol{r},\boldsymbol{s})} - a_{(\boldsymbol{r},\boldsymbol{s})} = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s})$ *for each* $\mathbf{s} \in \mathbb{N}^{m-k}$

iii) there exist a finite not empty subset $\mathcal{S}$ *of* $\mathbb{N}^{m-k}$ *such that, for each* $s \in \mathcal{S}$, $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}) < \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s})$, *hence* $a_{(\boldsymbol{r},\boldsymbol{s})} < b_{(\boldsymbol{r},\boldsymbol{s})}$.

Proof. From lemma 2.30 we have $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{v},\mathbf{s}) = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{v},\mathbf{s})\ \forall\ \mathbf{v} <_\mu \mathbf{r}$. Following proposition 2.23 we get $a_{(\boldsymbol{v},\boldsymbol{s})} = b_{(\boldsymbol{v},\boldsymbol{s})}\ \forall\ \mathbf{v} <_\mu \mathbf{r}$. From $a'_{\boldsymbol{r}} < b'_{\boldsymbol{r}}$ we get $\mathcal{H}^{\mathbf{u}}_{R/I}(\mathbf{r}) < \mathcal{H}^{\mathbf{u}}_{R/J}(\mathbf{r})$, hence

$$\sum_{\boldsymbol{s}\in\mathbb{N}^{m-k}} \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}) < \sum_{\boldsymbol{s}\in\mathbb{N}^{m-k}} \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s}). \tag{2.1}$$

Now $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s})$ and $\mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s})$ are zero for almost all $\mathbf{s} \in \mathbb{N}^{m-k}$ and inequality (2.1) implies that there exists at least one $\boldsymbol{\nu}$ such that $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\boldsymbol{\nu}) < \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\boldsymbol{\nu})$.
We define $\mathcal{S} := \{\mathbf{s} \in \mathbb{N}^{m-k} s.\ t. \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}) \neq \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s})\}$ and we have $\mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}) < \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s})$ for every $\mathbf{s} \in \mathcal{S}$. Therefore we have shown that i) and ii) imply iii). Let $\mathbf{s} \in \mathcal{S}$. As in the proof of proposition 2.23 we can write:

$$\sum_{(\boldsymbol{r},\boldsymbol{s})\in\mathbb{N}^m} (b_{(\boldsymbol{r},\boldsymbol{s})} - a_{(\boldsymbol{r},\boldsymbol{s})})\lambda^{(\boldsymbol{r},\boldsymbol{s})} = (1+f(\lambda)) \sum_{(\boldsymbol{r},\boldsymbol{s})\in\mathbb{N}^m} (\mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}))\lambda^{(\mathbf{r},\mathbf{s})}$$

where $f \in \mathbb{Z}[\lambda]$ and $\partial_{\mathbf{w}}(t) \neq 0$ for each $t \in \mathrm{Supp}\,(f)$, due to the fact that $\mathbf{u}$ is a system of weights. Then, from the above equality it follows

$$b_{(\boldsymbol{r},\boldsymbol{s})} - a_{(\boldsymbol{r},\boldsymbol{s})} = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\boldsymbol{s}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s}) + f(\lambda) \sum_{(\mathbf{v},\mathbf{t})\in\mathbb{N}^m} (\mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{v},\mathbf{t}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{v},\mathbf{t}))$$

with $\mathbf{v} <_\mu \mathbf{r}$. By assumption,

$$\mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{v},\mathbf{t}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{v},\mathbf{t}) = 0 \quad \text{if } \mathbf{v} <_\mu \mathbf{r}.$$

Therefore, for each $\mathbf{s} \in \mathbb{N}^{m-k}$ we have:

- $b_{(\boldsymbol{u},\boldsymbol{s})} = a_{(\boldsymbol{u},\boldsymbol{s})}$ if $\mathbf{u} <_\mu \mathbf{r}$
- $b_{(\boldsymbol{r},\boldsymbol{s})} - a_{(\boldsymbol{r},\boldsymbol{s})} = \mathcal{H}^{\mathbf{w}}_{R/J}(\mathbf{r},\mathbf{s}) - \mathcal{H}^{\mathbf{w}}_{R/I}(\mathbf{r},\mathbf{s})$

Hence i) and ii) are proved. □

Example 2.33 *Let* $R := k[x,y,z,t]$, $\mathbf{w} := \begin{bmatrix} 1 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 \end{bmatrix}$ *be a system of weights and* $\mathbf{u} := \begin{bmatrix} 1 & 1 & 2 & 1 \end{bmatrix}$ *a subsystem of weights of* $\mathbf{w}$. *The ideals* $I := (yt - t^2, xy - xt)$ *and* $J := (0) \subset I$ *are* $\mathbf{w}$*-homogeneous ideals of* R.
The Poincaré seriesof I *and* J *with respect to* $\mathbf{w}$ *and* $\mathbf{u}$ *are*

$$\mathcal{HP}^{\mathbf{w}}_{R/I}(\lambda_1,\lambda_2) = \frac{1 - \lambda_1^2\lambda_2^2 - \lambda_1^2\lambda_2^4 + \lambda_1^3\lambda_2^5}{(1-\lambda_1\lambda_2^3)(1-\lambda_1\lambda_2)^2(1-\lambda_1^2\lambda_2)}$$

$$\mathcal{HP}^{\mathbf{w}}_{R/J}(\lambda_1,\lambda_2) = \frac{1}{(1-\lambda_1\lambda_2^3)(1-\lambda_1\lambda_2)^2(1-\lambda_1^2\lambda_2)}$$

$$\mathcal{HP}^{\mathbf{u}}_{R/I}(\lambda) = \frac{1 - 2\lambda^2 + \lambda^3}{(1-\lambda)^3(1-\lambda^2))}$$

$$\mathcal{HP}^{\mathbf{u}}_{R/J}(\lambda) = \frac{1}{(1-\lambda)^3(1-\lambda^2))}$$

The first differences between I and J are generated by the terms yt and xy, of $\mathbf{w}$-degree $(2,2)$ and $(2,4)$ respectively, so we expect difference 2 in $\mathbf{u}$-degree 2 and differences 1 in $\mathbf{w}$-degrees $(2,2)$ and $(2,4)$.
By subtracting the numerators we compute the first differences and verify that this example fits with 2.23.

$$\langle J \rangle^{\mathbf{u}} - \langle I \rangle^{\mathbf{u}} = 2\lambda^2 + 7\lambda^3 + 17\lambda^4 + \cdots$$

$$\langle J \rangle^{\mathbf{w}} - \langle I \rangle^{\mathbf{w}} = \underbrace{\lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_2^4}_{\mathbf{u}-degree=2} + \lambda_1^3\lambda_2^3 + \lambda_1^3\lambda_2^5 + \lambda_1^3\lambda_2^7 + \lambda_1^4\lambda_2^3 + \cdots$$

Moreover, the reader can easily check how this example fits with lemma 2.31, and proposition 2.32.

Definition 2.34 *Let $\mathbf{u} = (w_1, \ldots, w_k)$ be a system of weights and $\mathbf{w} := (w_1, \ldots, w_m)$ be a refinement of $\mathbf{u}$. Let F be a $\mathbf{w}$-homogeneous system of generators of I and*

$$\langle I \rangle^{\mathbf{w}} = \sum_{\boldsymbol{d}\in\mathbb{N}^m} a_{\boldsymbol{u}}\lambda^{\boldsymbol{u}} \ , \ \langle \mathrm{Lt}_\sigma(F) \rangle^{\mathbf{w}} = \sum_{\boldsymbol{u}\in\mathbb{N}^m} b_{\boldsymbol{u}}\lambda^{\boldsymbol{u}}.$$

If $\mathbf{r} := Min(F, \mathbf{u})$, we denote $D(F, \mathbf{w}, \mathbf{u}) := \{(\mathbf{r}, \mathbf{s}) s.\ t. \mathbf{s} \in \mathbb{N}^{m-k}$ and $a_{(\boldsymbol{r},\boldsymbol{s})} \neq b_{(\boldsymbol{r},\boldsymbol{s})}\}$ and we call it the first difference set *of F with respect to $(\mathbf{w}, \mathbf{u})$.*

Theorem 2.35 *Let $\mathbf{w} = (w_1, \ldots, w_m)$ be a system of weights, I be a $\mathbf{w}$-homogeneous ideal, σ be a $\mathbf{w}$-compatible term-ordering and let G be the reduced Gröbner basis of I with respect to σ and F an interreduced $\mathbf{w}$-homogeneous system of generators of I. We assume*

$$\langle I \rangle^{\mathbf{w}} = \sum_{\boldsymbol{d}\in\mathbb{N}^m} a_{\boldsymbol{d}}\lambda^{\boldsymbol{d}} \ , \ \langle \mathrm{Lt}_\sigma(F) \rangle^{\mathbf{w}} = \sum_{\boldsymbol{d}\in\mathbb{N}^m} b_{\boldsymbol{d}}\lambda^{\boldsymbol{d}}$$

Finally we let $\mathbf{u}$ be a subsystem of weights of $\mathbf{w}$, $\mathbf{r} := Min_{\mathbf{w}}(F, \mathbf{u})$ and $D(F, \mathbf{w}, \mathbf{u})) = \{(r, u_1), \ldots, (r, u_s)\}$; then

i) $\#G_{(\boldsymbol{r},u_i)} = \#F_{(\boldsymbol{r},u_i)} + b_{(\boldsymbol{r},u_i)} - a_{(\boldsymbol{r},u_i)}$ *for $i = 1, \ldots, s$.*

ii) For each $h \in R_{(\boldsymbol{r},u_j)}$ and $i \neq j$ it holds $\boldsymbol{\mathcal{H}}_{R/\mathrm{Lt}_\sigma(F,h)}(\boldsymbol{r}, u_i) = \boldsymbol{\mathcal{H}}_{R/\mathrm{Lt}_\sigma(F)}(\boldsymbol{r}, u_i)$.

Proof. Let μ be the $\mathbf{w}$ degree ordering. We know that $\mathbf{r}$ is the first difference of F with respect to $\mathbf{u}$. Moreover F is assumed to be interreduced. This implies that F is an $\mathbf{r}'$-truncated reduced Gröbner basis of I for every $\mathbf{r}' <_\mu \mathbf{r}$. Therefore $F_{<_\mu\mathbf{r}} = G_{<_\mu\mathbf{r}}$. By Proposition 2.32

$$\dim\ _k(\mathrm{Lt}_\sigma(I)_{(\boldsymbol{r},u_i)}) - \dim\ _k(\mathrm{Lt}_\sigma(F)_{(\boldsymbol{r},u_i)}) = b_{(\boldsymbol{r},u_i)} - a_{(\boldsymbol{r},u_i)}$$

for each $1 \leq i \leq s$. From $F_{<_\mu\boldsymbol{r}} = G_{<_\mu\boldsymbol{r}}$ it follows that

$$\#G_{(\boldsymbol{r},u_i)} - \#F_{(\boldsymbol{r},u_i)} = \dim\ _k(\mathrm{Lt}_\sigma(I)_{(\boldsymbol{r},u_i)}) - dim_k(\mathrm{Lt}_\sigma(F)_{(\boldsymbol{r},u_i)})$$

for each $1 \leq i \leq s$, and so $i)$ is proved. To prove ii) we observe that if p,q are power products and $p \in R_{(\boldsymbol{r},u_i)}$, $q \in R_{(\boldsymbol{r},u_j)}$ then no divisibility relation can occur between them. $\square$

2.6.1 The Bi-Hilbert-Driven algorithm

Let $\mathbf{u} = (w_1, \dots, w_k)$ be a system of weights, $\mathbf{w} := (w_1, \dots, w_m)$ a refinement of $\mathbf{u}$ and $\mu \in \mathrm{TO}(\mathbb{Z}^m)$ compatible with $\mathbf{u}, \mathbf{w}$. Let I be a $\mathbf{w}$-homogeneous ideal of $R := k[x_1, \dots, x_n]$, $G := \{g_1, \dots, g_r\}$ a set of $\mathbf{w}$-homogeneous generators of I, $\mathbf{w}$-Poincaré seriesof R/I and σ a $\mathbf{w}$-compatible term-ordering. We want to compute the Gröbner basis of I with respect to σ, using the Poincaré series.

Remark 2.36 *Since the bigger the* $\mathbf{u}$ *homogeneous components the better the splitting, it is useful to take* $\mathbf{u}$ *of type one. If this is not the case, en lieu of* $\mathbf{u}, \mathbf{w}$ *we can use* $\mathbf{u}', \mathbf{w}'$*, where*

1. $\mathbf{u}' := (w_1')$*, where* $u_1 := \sum_{i=1}^{k} w_i$ *and*
2. $\mathbf{w}' := (w_2, \dots, w_m)$.

The equivalence of $\mathbf{u}, \mathbf{u}'$ *and* $\mathbf{w}, \mathbf{w}'$ *is straightforward.*

Example 2.37 *Let us consider the system of weights* $\mathbf{w} := \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$

Of course no subset is a subsystem of weights, but if an ideal is $\mathbf{w}$*-homogeneous, it is also homogeneous with respect to* $\mathbf{w}' := \begin{bmatrix} 1 & 1 & 3 & 2 \\ 1 & 0 & 2 & 1 \end{bmatrix}$

So it is possible to consider the subsystem of weights given by $\mathbf{u} := \begin{bmatrix} 1 & 1 & 3 & 2 \end{bmatrix}$.

Input $G := \{g_1, \dots, g_r\} \subseteq k[\mathbf{x}]$, with g_i homogeneous;
$\tau \in \mathrm{TO}(R)$;
$\mathbf{u}$ system of weights and $\mathbf{w}$, refinement of $\mathbf{u}$;
$\langle (G) \rangle^{\mathbf{w}}$;
$\langle (G) \rangle^{\mathbf{u}}$;
Output the reduced Gröbner basis of G with respect to τ.

begin
  $L :=$ generates the initial pairs from G;
  $d' := \mathrm{in}(\langle \mathrm{Lt}_\tau(G) \rangle^{\mathbf{u}}) - \langle (G) \rangle^{\mathbf{u}}$;
  $\mathbf{d} := \mathrm{in}(\langle \mathrm{Lt}_\tau(G) \rangle^{\mathbf{w},d'} - \langle (G) \rangle^{\mathbf{w}})$;
  D :=list of the $\mathbf{w}$ degrees of the terms in $\mathbf{d}$;
  $d_h := Head(D)$; $D := Tail(D)$
  Delete the pairs in L whose $\mathbf{u}$ degree is less than d';
  Delete the pairs in L whose $\mathbf{u}$ degree is equal to d' and whose $\mathbf{w}$ degrees are not in $\mathbf{d}$.
  while $L \neq \emptyset$ **do begin**
    choose $(g_i, g_j) \in L$ s.t. $d_h = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
    $L := L - \{(g_i, g_j)\}$;
    $g := \mathrm{NF}\,_\tau^{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j), G)$;
    if $f \neq 0$ **then** $G' := G' \cup \{g\}$;
    if $d_h \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
      begin
        if $d' = min_{\mathbf{u}}\{\partial_{\mathbf{u}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
          begin
            $d_h := Head(D)$; $D := Tail(D)$
            $G' := Interreduce(G')$;
            $L, G := \mathtt{Upgrade}^{\mathbf{w},\mathbf{d}}(L, G', G)$;
            $L' := \emptyset$; $G' := \emptyset$;
          end

else
begin
$d' := \mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w}}) - \langle (G)\rangle^{\mathbf{u}}$
$\mathbf{d} := \mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w},d'} - \langle (G)\rangle^{\mathbf{w}})$
D :=list of the $\mathbf{w}$ degrees of the terms in $\mathbf{d}$;
Delete the pairs in L whose $\mathbf{u}$ degree is less than d';
Delete the pairs in L whose $\mathbf{u}$ degree is equal to d' and whose $\mathbf{w}$ degrees are not in $\mathbf{d}$.
$d_h := Head(D)$; $D := Tail(D)$
end
endif;
endwhile;
return G;
end.

Remark 2.38

- *Following the previous remark, Bi-Hilbert-Driven requires the computation of some univariate Hilbert-Poincaré series in order to spot the first difference, say in* $\mathbf{u}$*-degree* r*. Once such a difference is found, Bi-Hilbert-Driven requires to compute a multigraded Hilbert-Poincaré series in order to detect the differences, say* $((r,u_1),\dots,(r,u_s))$*, in the more refined graduation.*

- *Once a first difference is found, say in* $\mathbf{u}$*-degree* r*, we do not need to compute the full multigraded Hilbert-Poincaré series, but it is sufficient to truncate it to the* $\mathbf{u}$*-degree* r*. This is a substantial optimization of the Bi-Hilbert-Driven algorithm.*

- *The computation of the SP-polynomials in the degrees* $((r,u_1),\dots,(r,u_s))$ *is intrinsically parallelizable. Namely,* $R_{(r,u_1)} + \cdots + R_{(r,u_s)}$ *is a direct sum, hence no interaction occurs among the elements of different subdegrees. This allows us to conclude that the computations in the degrees* $(r,u_1),\dots,(r,u_s)$ *are independent each other.*

2.6.2 Some statistics for the Hilbert-Driven and Bi-Hilbert-Driven algorithms

We examine here four examples for the Hilbert-Driven and Bi-Hilbert-Driven Algorithms. For more examples, see [T, CDR]. In every case, the Poincaré series is known because the ideals are complete intersections.

Example 2.39 *Let* $R := \mathbb{Z}_{32003}[x_1,\dots,x_3,y_1,\dots,y_3]$*,*

$$I := (x_1y_1^2 - x_2y_2^2 + x_3y_3^2, x_1y_2^3 - x_2y_3^3 + x_3y_1^3, x_1y_3^4 - x_2y_1^4 + x_3y_2^4)$$

$$\mathbf{w} := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

and τ *be the term-ordering given by competing the matrix* $\mathbf{w}$ *with* degrevlex*. We remark that* I *is homogeneous with respect to the standard graduation and* $\mathbf{w}$*. We want to compute the Gröbner basis of* I *with respect to* τ*.*

Example 2.40 *Let* $R := \mathbb{Z}_{32003}[y_1, \dots, y_{36}, x_0, \dots, x_9]$,

$$
\begin{aligned}
I := (&x_1^2 - x_0x_2 + y_1, x_1x_2 - x_0x_3 + y_2, x_1x_3 - x_0x_4 + y_3, x_1x_4 - x_0x_5 + y_4\\
&x_1x_5 - x_0x_6 + y_5, x_1x_6 - x_0x_7 + y_6, x_1x_7 - x_0x_8 + y_7, x_1x_8 - x_0x_9 + y_8\\
&x_2^2 - x_1x_3 + y_9, x_2x_3 - x_1x_4 + y_{1-}, x_2x_4 - x_1x_5 + y_{11}, x_2x_5 - x_1x_6 + y_{12}\\
&x_2x_6 - x_1x_7 + y_{13}, x_2x_7 - x_1x_8 + y_{14}, x_2x_8 - x_1x_9 + y_{15}, x_3^2 - x_2x_4 + y_{16}\\
&x_3x_4 - x_2x_5 + y_{17}, x_3x_5 - x_2x_6 + y_{18}, x_3x_6 - x_2x_7 + y_{19}, x_3x_7 - x_2x_8 + y_{20}\\
&x_3x_8 - x_2x_9 + y_{21}, x_4^2 - x_3x_5 + y_{22}, x_4x_5 - x_3x_6 + y_{23}, x_4x_6 - x_3x_7 + y_{24}\\
&x_4x_7 - x_3x_8 + y_{25}, x_4x_8 - x_3x_9 + y_{26}, x_5^2 - x_4x_6 + y_{27}, x_5x_6 - x_4x_7 + y_{28}\\
&x_5x_7 - x_4x_8 + y_{29}, x_5x_8 - x_4x_9 + y_{30}, x_6^2 - x_5x_7 + y_{31}, x_6x_7 - x_5x_8 + y_{32}\\
&x_6x_8 - x_5x_9 + y_{33}, x_7^2 - x_6x_8 + y_{34}, x_7x_8 - x_6x_9 + y_{35}, x_8^2 - x_7x_9 + y_{36})
\end{aligned}
$$

$$
\mathbf{w} := \left[\begin{array}{cccccccccccccccccccccccccccccccccccccccccccccc}
2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&2&1&1&1&1&1&1&1&1&1&1\\
2&3&4&5&6&7&8&9&4&5&6&7&8&9&10&6&7&8&9&10&11&8&9&10&11&12&10&11&12&13&12&13&14&14&15&16&0&1&2&3&4&5&6&7&8&9
\end{array}\right]
$$

and let τ *be the term-ordering given by completing the matrix* $\mathbf{w}$ *with the elimination term-ordering for the variables* $\mathbf{x}$. *We remark that* I *is homogeneous with respect to the graduations given by* w_1 *and* $\mathbf{w}$. *We want to compute* $\mathrm{GB}(I, \tau)$.

Example 2.41 *Let* $R := \mathbb{Z}_{32003}[y_1, \dots, y_{16}, x_{1,1}, \dots, x_{4,4}, t]$ *and*

$$
M := \left[\begin{array}{cccc}
x_{1,1} + x_{1,2} + x_{2,1} & x_{1,2} & x_{1,3} & x_{1,4}\\
x_{2,1} & x_{2,2} + x_{2,4} + x_{1,3} & x_{2,3} & x_{2,4}\\
x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4}\\
x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4}
\end{array}\right]
$$

Then let I *be the ideal generated by the polynomials* $\{tf_j - y_j | f_j \in Min_3(M)$ *and* $j : 1 \dots 16\}$ *and* $\mathbf{w}$ *the system of weights*

$$
\left[\begin{array}{ccccccccccccccccccccccccccccccccc}
4&4&4&4&4&4&4&4&4&4&4&4&4&4&4&4&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
9&9&9&9&10&10&10&10&12&12&12&12&12&12&12&12&2&2&2&2&2&2&2&2&4&4&4&4&5&5&5&5&1
\end{array}\right]
$$

Let τ *the elimination term-ordering for the* t *variable. Remark that* I *is homogeneous with respect to the system of weights given by* w_1 *and* $\mathbf{w}$. *We want to compute* $GB(I, \tau)$.

Example 2.42 *Let* $R := \mathbb{Z}_{32003}[t, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, u]$,
$\tau := \mathrm{lex}(t > \dots > u)$,

$$
\begin{aligned}
I := (&-tc^2e + 2tbcf - taf^2 - tb^2h + taeh + k^4,\\
&- tcde + tbdf + tbcg - tafg - tb^2i + taei + l^4,\\
&- td^2e + 2tbdg - tag^2 - tb^2j + taej + m^4,\\
&- tcdf + tc^2g + tbdh - tagh - tbci + tafi + n^4,\\
&- td^2f + tcdg + tbdi - tagi - tbcj + tafj + o^4,\\
&- tdf^2 + tcfg + tdeh - tbgh - tcei + tbfi + p^4,\\
&- tdfg + tcg^2 + tdei - tbgi - tcej + tbfj + q^4,\\
&- td^2h + 2tcdi - tai^2 - tc^2j + tahj + r^4,\\
&- tdgh + tdfi + tcgi - tbi^2 - tcfj + tbhj + s^4,\\
&- tg^2h + 2tfgi - tei^2 - tf^2j + tehj + u^4)
\end{aligned}
$$

and

$$
\mathbf{w} := \left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
4&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1
\end{array}\right]
$$

Remark that I *is homogeneous with respect to the systems of weights given by* $\mathbf{w}_1$ *and* $\mathbf{w}$. *We want to to compute* $GB(I, \tau)$.

We show two tables. In the first one we report the timing in seconds obtained with the above described examples. In the second table we show some statistics related to the performance of our algorithm. We remark that in all the examples the coefficient filed is $\mathbb{Z}_{32003}$. In $\mathbb{Q}$ the performance of the algorithm improves, since the computational cost of every reduction increases.

And now let us give some explanation of the headings:

Usual The usual Buchberger algorithm.

HD The univariate Hilbert-Driven Algorithm.

Ov$_{\mathbf{HD}}$ The overhead of the computation of the univariate Poincarè series.

BHD The Bi-Hilbert-Driven algorithm.

Ov$_{\mathbf{BHD}}$ The overhead of the computation of the univariate and the multigraded Poincarè series.

Useful The number of the computed useful pairs after the application of the Gebauer-Möller criteria.

Total The total number of the computed critical pairs.

HD-K The number of pairs recognized as useless (hence killed) by the Hilbert-Driven algorithm.

BHD-K The number of pairs recognized as useless (hence killed) by the Bi-Hilbert-Driven algorithm.

 # PS The number of univariate and multigraded Poincarè series computed.

Name	Usual	HD	Ov$_{\mathbf{HD}}$	BHD	Ov$_{\mathbf{BHD}}$
Ex 2.39	106 s	94 s	0.16 s	10 s	2.5 s
Ex 2.40	13611 s	772 s	0.38 s	556 s	8.08 s
Ex 2.41	9895 s	563 s	3.84 s	333 s	79.55s
Ex 2.42	26 s	16 s	0.76 s	15 s	4.21 s

Name	Useful	Total	HD-K	BHD-K	# PS
Ex 2.39	272	1073	89	711	28+28
Ex 2.40	1659	53697	35728	46700	8+8
Ex 2.41	820	12097	2587	7398	24+24
Ex 2.42	411	4123	1524	1708	25+25

Remark 2.43

The BHD algorithm kills more pairs than the HD algorithm.

The computation of Poincaré series for system of weights of type > 1 is quite high with respect to the cost of the computation of Poincaré series for type one system of weights, that is negligible.

The BHD algorithm has better timings notwithstanding the cost of high type Poincaré series computations.

Remark 2.44 *If the memory available is not sufficient to avoid swapping, the BHD Algorithm is even more interesting, since the cost of each reduction is bigger. We see the timings for a Pentium 90 with 32MB RAM.*

Name	Usual	HD	BHD
Ex 2.39	359 s	289 s	34 s
Ex 2.40	$>$ 24h	16908 s	4202 s
Ex 2.41	>16000 s	2510 s	1465 s
Ex 2.42	96 s	52 s	48 s

2.7 Dynamic Buchberger algorithm

The computation of a Gröbner basis may have very different costs depending on the term-ordering chosen. If there are no restrictions on the choice of term-ordering, a good heuristic strategy is to examine the input and search by hand a suitable term-ordering for the computation.

Example 2.45 *Let* $R := \mathbb{Q}[a,b,c,d,e,f,g,h]$ *and*

$$\begin{aligned} C_7 := Ideal(\ & a+b+c+d+e+f+g, \\ & ab+bc+cd+de+ef+fg+ga, \\ & abc+bcd+cde+def+efg+fga+gab, \\ & abcd+bcde+cdef+defg+efga+fgab+gabc, \\ & abcde+bcdef+cdefg+defga+efgab+fgabc+gabcd, \\ & abcdef+bcdefg+cdefga+defgab+efgabc+fgabcd+gabcde, \\ & h^7-abcdefg\ ) \end{aligned}$$

Then the time for the computation of the reduced Gröbner basis with respect to the degrevlex$(a > \cdots > h)$ *is* 306", *there are* 443 *polynomials in the basis and* 79897 *terms in the supports of the basis polynomials. If the variable* h *is bigger than the others, the last equation is neutralized, hence a good choice seems to be to use* degrevlex$(h > a > \cdots > g)$. *Indeed, the computation time for the this ordering is* $62''$, *there are* 121 *polynomials in the basis and* 28595 *terms in the supports of the basis polynomials.*

Such a priori choice is, apart from some cases, usually quite difficult. Two well known useful heuristic criteria are :

- lexicographically term-orderings are usually bad, and the degrevlex term-orderings are usually good;
- it is generally a good idea to use a system of weights on the ring that makes the ideal of which we want to compute the Gröbner basis homogeneous, or as homogeneous as possible.

The following theorem, due to Mora and Robbiano, [MR], suggests that an "optimal" term-ordering exists.

Theorem 2.46 *Let* $R := k[x_1, \ldots, x_n]$ *and* I *an ideal of* R. *Then* $\mathrm{TO}(R)$ *is partitioned in a finite number of subsets such that the term-orderings in the same subset give the same reduce Gröbner basis.*

Proof. See [MR]. □

Theorem 2.46 suggests to find constructively the "cheapest" reduced Gröbner basis for a given ideal by the choice of a suitable term-ordering. It is in general impossible to choose this term-ordering a priori, since the Buchberger algorithm is quite impredictable; the ordering can, however, be adapted dynamically during the execution of the Buchberger algorithm, exploiting the data made available during the computation.

It is thus of interest to develop a version of the Buchberger algorithm that choose, during the computation, the best possible term-ordering, according to some heuristics. See [Ca]

2.7.1 Gröbner Fan

We use the notation of section 1.2.3.

Definition 2.47 *Let* $R := k[x_1, \ldots, x_n]$ *and* $F := \{f_1, \ldots, f_k\} \subset R$. *For each* f_i *let us chose* $m_i \in \mathrm{Supp}\,(f_i)$. *Let* $\mathrm{PL}\,(F)$ *be the list of such terms. The elements of* $\mathrm{PL}\,(F)$ *are* potential leading terms *of* F *if there exists* $\sigma \in \mathrm{TO}(R)$ *such that* $\mathrm{Lt}_\sigma(f_i) = m_i\ \forall\ i : 1 \ldots k$. *The list* $\mathrm{PL}\,(F)$ *is a* list of potential leading terms *of* F *and* σ *is* compatible with $F, \mathrm{PL}\,(F)$. *Let* $f \in R$ *and* $m \in \mathrm{Supp}\,(f)$. *Then* m *is a* potential leading term of f with respect to $F, \mathrm{PL}\,(F)$ *if* $\mathrm{PL}\,(F) \cup \{m\}$ *is a list of possible leading terms of* $F \cup \{f\}$.

Remark 2.48 *Let $R := k[x_1, \ldots, x_n]$, $\tau, \sigma \in \mathrm{TO}(R)$ and $F := \{f_1, \ldots, f_k\} \subset R$.*

1. *If $\mathrm{PL}(F)$ is a list of potential leading terms of F, then there exists at least one potential leading term of f with respect to $F, \mathrm{PL}(F)$, e. g. $\mathrm{Lt}_\sigma(f)$, where σ is a term-ordering compatible with $F, \mathrm{PL}(F)$.*

2. *Then if τ is compatible with $F, \mathrm{Lt}_\sigma(F)$, then τ is equivalent to σ with respect to F, e. g. τ and σ gives to F the same leading terms.*

Theorem 2.49 *Let $R := k[x_1, \ldots, x_n]$ and $F := \{f_1, \ldots, f_k\} \subset R$. Let $\mathrm{PL}(F)$ be a list of potential leading terms of F and $f \in R$. Then the relation $\sim_{F \cup \{f\}}$ induces a partition of the set of term-orderings compatible with $F, \mathrm{PL}(F)$ in equivalence classes such that*

1. *every class is composed of the term-orderings equivalent with respect to $F \cup \{f\}$;*

2. *the cones associated to the classes are polyhedral cones, centered on the origin, pairwise disjoint in $(\mathbb{R}^n)^+$ and with topological interior non void*

Proof. Follows immediately from proposition 1.74 and corollary 1.76. □

Corollary 2.50 *Every equivalence class has a representative whose first vector is in $(\mathbb{N}^n)^+$*

Corollary 2.51 *There is a one-to-one correspondence between potential leading terms of f and equivalence classes (and their associated cones). The number of classes is then finite, and $\leq \# \mathrm{Supp}(f)$. Thereafter such correspondence will be implicit.*

Definition 2.52 *Under the hypothesis of the theorem 2.49, to every potential leading term t of f with respect to $F, \mathrm{PL}(F)$ is associated a class, $\underline{\sigma : t}$. If t belongs to $\mathrm{Supp}(f)$ but it is not a potential leading term, then $\underline{\sigma : t}$ is the empty class.*

Definition 2.53 *Let $R := k[x_1, \ldots, x_n]$, $F := \{f_1, \ldots, f_k\} \subset R$ and $f \in R$. Let $\mathrm{PL}(F)$ be a list of potential leading terms for F. Then the set of cones associated to the equivalence classes with respect to $\mathrm{PL}(F), f$ is the* Gröbner fan of F with respect to $\mathrm{PL}(F), f$.

2.7.2 The Dynamic Buchberger algorithm

Proposition 2.54 *Let $R := k[x_1, \ldots, x_n]$, $F := \{f_1, \ldots, f_k\} \subset R$ and $\tau, \sigma \in \mathrm{TO}(R)$ such that $\tau \sim_F \sigma$. Then $\mathrm{NF}_{F,\tau}(f) = \mathrm{NF}_{F,\sigma}(f)$ for every $f \in R$.*

Proof. Follows immediately from the definitions of NF and $\sim_F$. □

Remark 2.55 *From proposition 2.54 it follows that to exploit the reduction already performed, it is sufficient for the new term-ordering to be equivalent to the old one with respect to the already computed polynomials.*

We give a quick review of the Dynamic Buchberger Algorithm :
Initially, we select a suitable term-ordering through considerations on the input polynomials. Then, we execute the Buchberger algorithm but, after every S-Polynomial reduction which doesn't give zero, we choose a leading term among the potential leading terms of the reduced polynomial with respect to the current term-ordering and the already computed polynomials. To this term is associated a term-ordering, equivalent to the current term-ordering with respect to the already computed polynomials. This is the new term-ordering used thereafter by the algorithm. The choice is directed by heuristics based on the Hilbert function (see subsection 2.7.3).

Dynamic Buchberger algorithm

Input $G := \{g_1, \dots, g_r\} \subseteq k[\mathbf{x}]$ homogeneous, $\mathbf{w}$ system of weights for R.
Output the reduced Gröbner basis of G with respect to τ;.

begin
  L:= generates the initial pairs;
  $\tau :=$ `ComputeInitialOrder`(G);
  d:=$min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$;
  while $L \neq \emptyset$ **do begin**
    choose $(g_i, g_j) \in L$ s.t. $d = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
    $L := L - \{(g_i, g_j)\}$;
    $g := \mathrm{NF}_{\sigma}(\mathrm{SPoly}\,(g_i, g_j), G)$;
    if $g \neq 0$ **then** G':=$G' \cup \{g\}$;
    if $d \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
      begin
        $\tau :=$ `RefineCurrentOrder`(G, g, τ);
        G':=$Interreduce(G')$;
        L,G:=`Upgrade`(L, G', G);
        L':=$\emptyset$;
        d:=$min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$;
      endif;
  endwhile;
  return G;
end.

The only differences of the Dynamic Buchberger Algorithm with respect to the usual Buchberger algorithm is the repeated change of term-ordering during the execution, controlled by the procedures `ComputeInitialOrder` and `RefineCurrentOrder`.

2.7.3 Order refining

Procedure RefineCurrentOrder

Input$F := \{f_1, \dots, f_k\} \subseteq R$, $f \in R$, $\sigma \in \mathrm{TO}(R)$;
Output$\tau \in \mathrm{TO}(R)$, compatible with σ, F;

begin
  Step a Determine all potential leading terms of f with respect to $F, \mathrm{Lt}_{\sigma}(F)$;
  Step b Choose a term t among the terms produced by step **a**;
  return τ, a term-ordering belonging to the class associated to t with respect to $F, \mathrm{Lt}_{\sigma}(F)$;
end.

Step a

To determine all the potential leading terms of f with respect to $F, \mathrm{Lt}_{\sigma}(F)$ we have to find integral solutions for the associated systems of linear inequalities (test systems) as discussed in subsection 1.2.3. One of the methods for solving a system of rational inequalities if the simplex algorithm. Let S_t be such a system.

Remark 2.56 *If a test system S_t has been solved by the simplex algorithm, a system obtained adjoining some condition to S_t is usually easily solvable by the simplex algorithm starting from the solution of system S_t. Exploiting this fact, an incremental version of the simplex algorithm has been developed, thus cutting the overhead significantly.*

Some easy optimizations greatly reduce the number of test systems that have to be considered.

Remark 2.57 *If a term divides another term, then it is not a potential leading term.*

Proposition 2.58 *Let $R := k[x_1, \dots, x_n]$, $F := \{f_1, \dots, f_k\} \subset R$ and $(t_1, \dots, t_k)$ be potential leading terms of $f_1, \dots, f_k$. Then the system*

$$\left\{ \; t_i > m \quad m \in \operatorname{Supp}(f_i), \; i : 1 \dots k \right. \tag{2.2}$$

is equivalent to the system

$$\left\{ \; t_i > m \quad m \textit{ potential leading term of } f_i \textit{ with respect to } \{f_1, \dots, f_{i-1}\}, \{t_1, \dots, t_{i-1}\}, \; i : 1 \dots k \right. \tag{2.3}$$

Proof. Let $F' := \{f_1, \dots, f_{i-1}\}$ and $\operatorname{Lt}(F') := \{t_1, \dots, t_{i-1}\}$. To prove the thesis it is sufficient to show that every condition $t_i > m$ in 2.2 such that m is not a potential leading term of f_i is useless. It is hence sufficient to show that for every term-ordering $\sigma \in \operatorname{TO}(R)$ compatible with $F', \operatorname{Lt}(F')$ there exists t_σ potential leading term of f_i with respect to $F', \operatorname{Lt}(F')$ such that $t_\sigma > m$. From that it follows that $t_i > t_\sigma \Longrightarrow t_i > m$ and then, since the first condition is in system 2.2, the second is useless. But then as t_σ it suffices to choose $\operatorname{Lt}_\sigma(f_i)$. $\square$

Remark 2.59 *Remark 2.57 allows us to skip some candidates to potential leading terms position. The use of proposition 2.58 is a bit more delicate. We have to keep trace of the potential leading terms of every computed polynomial, and only these terms are used to generate the test systems. As the number of reduction performed increases, the polyhedral cones associated to the test systems narrows, and so the number of potential leading terms for every polynomial usually decreases substantially after the first reductions.*

Remark 2.60 *A further optimization in the description of the cones associate to the test systems may be done using convex geometry techniques en lieu of the simplex, see [Sw], obtaining a minimal description of the cones. Such techniques are, however, more expensive in time and space than the simplex algorithm, at least for examples of the size examined in this thesis.*

Step b

We have the set S of potential leading terms of the poly f with respect to $F, \operatorname{Lt}_\sigma(F)$. We want to choose a leading term in S. The heuristics we use is based on Hilbert function considerations, and we consider only homogeneous ideals.

Definition 2.61 *Let f and g be two functions from $\mathbb{N}^n$ to $\mathbb{N}$ and $\tau \in O(\mathbb{N}^n)$. Then f is* lexicographically greater *than g with respect to τ ($f \geq_{\text{lex},\tau} g$) if the first (with respect to τ) $\mathbf{d} \in \mathbb{N}^n$ for which $f(\mathbf{d}) \neq g(\mathbf{d})$ is such that $f(\mathbf{d}) > g(\mathbf{d})$.*

Proposition 2.62 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{w}$ be a system of weights for R of type r, $\tau \in \operatorname{TO}(\mathbb{N}^r)$, $F := \{f_1, \dots, f_k\}$ be homogeneous polynomial in R and $I := (F)$. Then*

a) $\boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/\operatorname{Lt}_\sigma(F)} \geq_{\text{lex},\tau} \boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/I}$ for every $\sigma \in \operatorname{TO}(R)$.

b) Equality holds if and only if F is a Gröbner basis of I with respect to σ.

Proof.

a) From the definition of Hilbert function follows that $\boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/\operatorname{Lt}_\sigma(F)}(\mathbf{d}) \geq \boldsymbol{\mathcal{H}}^{\mathbf{w}}_{R/I}(\mathbf{d}) \; \forall \; \mathbf{d} \in \mathbb{N}^r$.

b) If F is a Gröbner basis of I, then $Lt_\sigma(I) = Lt_\sigma(F)$, hence $\mathcal{H}^{\mathbf{w}}_{R/\operatorname{Lt}_\sigma(F)} = \mathcal{H}^{\mathbf{w}}_{R/I}$.
If $\mathcal{H}^{\mathbf{w}}_{R/\operatorname{Lt}_\sigma(F)} = \mathcal{H}^{\mathbf{w}}_{R/I}$, assume that F is not a Gröbner basis. Then there exists a term $t \in \operatorname{Lt}_\sigma(I) - \operatorname{Lt}_\sigma(F)$. Since $\operatorname{Lt}_\sigma(I) \supset \operatorname{Lt}_\sigma(F)$, such term makes the Hilbert function of $\operatorname{Lt}_\sigma(I)$ different from the Hilbert function of $\operatorname{Lt}_\sigma(F)$. □

Keeping in mind proposition 2.62, a reasonable heuristics is :
a potential leading term t is considered better than another potential leading term t' with respect to F, σ if the Hilbert function of $\operatorname{Lt}_\sigma(F)\cup\{t\}$ is lexicographically lesser than the Hilbert function of $\operatorname{Lt}_\sigma(F)\cup\{m\}$.

Proposition 2.63 *Let $R := k[x_1, \ldots, x_n]$, $\mathbf{w}$ be a system of weights for R of type r, $\tau \in \operatorname{TO}(\mathbb{N}^r)$, I be a monomial ideal of R and $t, t' \in T^n$. Then*

a) If $\partial_{\mathbf{w}}(t) > \partial_{\mathbf{w}}(t')$, then $\mathcal{H}^{\mathbf{w}}_{R/(I,t)} \geq_{\operatorname{lex},\tau} \mathcal{H}^{\mathbf{w}}_{R/(I,t')}$.

b) If $\partial_{\mathbf{w}}(t) = \partial_{\mathbf{w}}(t')$, then $\mathcal{H}^{\mathbf{w}}_{R/(I,t)} \geq_{\operatorname{lex},\tau} \mathcal{H}^{\mathbf{w}}_{R/(I,t')} \iff \mathcal{H}^{\mathbf{w}}_{R/(I:(t))} \leq_{\operatorname{lex},\tau} \mathcal{H}^{\mathbf{w}}_{R/(I:(t'))}$.

c) $\mathcal{H}^{\mathbf{w}}_{R/(I,t)} \geq_{\operatorname{lex},\tau} \mathcal{H}^{\mathbf{w}}_{R/(I,t')} \iff$ the first (w.r.t. τ) term of $\langle I,t\rangle^{\mathbf{w}} - \langle I,t'\rangle^{\mathbf{w}}$ is positive.

Proof. The proof of *a)* is immediate and *c)* is a rewriting of proposition 2.32. The proof of *b)* follows from the main formula 1.110. □

Using proposition 2.63, an implementation of step **b** for **w** homogeneous ideals is the following algorithm:

Input $L := \{t_1, \ldots, t_k\} \subset T^n$, **w** system of weights, I monomial ideal of R;
Output $t \in T^n$;

begin
Discard from L all terms except the ones with minimum **w** degree;
Discard from L all terms except the term t for which $\mathcal{H}^{\mathbf{w}}_{R/(I:t)}((I,t))$ is lexicographically maximal;
return a random element of L.
end.

If the ideal is not homogeneous, one may use other heuristics, or even the Hilbert function heuristics on the ideal of leading terms. In practice, for non-homogeneous ideals, the Hilbert function heuristics on the ideal of leading terms works as well as for homogeneous ideals. Another choice is to use the ordering that, interpreted as system of weights, gives the maximum of homogeneity to the ideal. This last choice has not yet been tested.

2.7.4 Initial orderings

The choice of the initial term-ordering is very important, since it sets the initial cone, that will be further refined, but inside which the orders will remain throughout the algorithm. hence, even if it is possible to use the steps **a** and **b** in the same way as in the procedure `Refine CurrentOrder`, some improvement with respect to that procedure are useful.

A simple but effective approach is to consider all the sets of possible term-orderings of the whole initial list of polynomials, and then to choose the one with the minimal Hilbert function. The disadvantage of this method is that the cardinality of the sets grows very fast with the complexity of the generators of the input list. It is in general useful not to consider all possible term-orderings, but only the most promising.

Input $F := \{f_1, \ldots, f_k\} \subset k[x_1, \ldots, x_n] =: R$;
Output $\sigma \in \operatorname{TO}(R)$;

begin

1. Create the list L of all the possible term-orderings for the first p, $p \leq k$ polynomials in the input, p being the highest possible compatible with the available computational power;
2. Discard from L the elements with non minimal lexicographically Hilbert function.;

For every polynomial f_i, $i : p \dots k$.

a. the potential leading terms of f_i with respect to every term-ordering in L are found.
b. Every element of L is substituted with a compatible term-ordering for each potential leading term of f_i.
c. Discard from L the term-orderings with non lexicographically minimal Hilbert function.

return one element of L.

end.

The algorithm sketched above may be implemented and optimized using the techniques already shown during the discussion of the procedure `RefineCurrentOrder`.

Remark 2.64 *Usually, a polynomial set would be interreduced before the computation of the Gröbner basis. In the dynamic case, such interreduction is meaningless, since at start there is no term-ordering. To obtain the same effect, in step* 3*, at the* i^{th} *iteration,* f_i *may be reduced with respect to* $\{f_1, \dots, f_{i-1}\}$ *and each term-ordering in L, and the subsequent computation may be performed using these reduced forms.*

Remark 2.65 *If we put some constraint to the dynamic ordering that we are building, the Dynamic Algorithm works in the same way. the only difference is that, in step* **a, the system that are being solved contain the constraint. It is thus possible to use the Dynamic Algorithm for elimination ordering, degree-compatible ordering and so on.**

An experimental version of the Dynamic Algorithm has been implemented in CoCoA 1.0, and some examples executed on a Macintosh LC. From this experience, it is possible to draw some conclusions

- The further the computation proceeds, the lesser the potential leading terms for every polynomial.
- The overhead due to the simplex algorithm is hence reasonable.
- The heuristics employed are not stable, in the sense that minimizing the Hilbert function step by step gives no guarantees of minimizing the Hilbert function over several steps.
- The term-ordering selected by the algorithm leads to simpler Gröbner basis (less polynomial, less degrees) than usual term-orderings. The number of pairs processed is also usually small. The coefficients often grow more than usual, sometimes explosively.
- The timings are inconclusive for the examples examined, except for binomial ideals, where the timings are quite good.

From this limited data, it is possible to conclude that the Dynamic Algorithm may be interesting in general for binomial examples, where the coefficient growth doesn't happen, for examples with relatively few, quite dense polynomials, where the overhead is insignificant, and where it is of interest keeping the Gröbner basis small.

Chapter 3

Syzygies and resolutions

In this chapter we examine some algorithm related to the syzygy computation, especially in the homogeneous case. The chapter is so structured :

minimal basis algorithm for homogeneous modules, section 3.1;

syzygy computation, section 3.2 and

resolution computation, sections 3.3-3.7.

3.1 Minimal basis

Definition 3.1 *Let $R := k[x_1, \dots, x_n]$ and $M \subset R^r$ be an R module. Let $L := (m_1, \dots, m_k)$ be a system of generators for M. Then L is a* minimal system of generators *or a* minimal basis *for M if $L - \{f_i\}$ doesn't generate M for any $i : 1 \dots k$.*

Proposition 3.2 *Let $R := k[x_1, \dots, x_n]$ and $M \subset R^r$ be a $\mathbf{w}$ homogeneous R module. Then all the minimal system of generators of M have the same number of elements.*

Proof. Immediate from the Hilbert Function theory, ref. section 1.4. □

Proposition 3.3 *With the notation of definition 3.1, let G be the reduced Gröbner basis of M with respect to a module term-ordering μ and L be a reduced system of generators of M. Then the polynomials that belong to both the lists L and G are a minimal basis of M.*

Proof. Immediate. □

The construction of a minimal basis of an ideal/module (and the minimum number of generators) are relevant questions in themselves. From the computational point of view, the construction of a minimal bases may be interesting if further computations are required on the ideal e. g. intersection, resolution etc. etc.

From proposition 3.3, it is easy to deduce an easy algorithm to compute the minimal basis of M: we compute the Gröbner basis of (L), truncated to the maximum $\mathbf{w}$ degree of L, and then choose the elements in L which belong to the building basis. they are a minimal basis for M by proposition 3.3. Since it would be useful during the discussion of the computation of the minimal resolution, a version of the homogeneous Buchberger algorithm optimized for the computation of the minimal basis is shown below.

Input $G := \{g_1, \dots, g_r\} \subseteq k[\mathbf{x}]$, with g_i $\mathbf{w}$ homogeneous
$\tau \in \mathrm{TO}(R)$
$\mathbf{w}$, system of weights for R
Output a minimal Gröbner basis of G.

begin
  $F := G$;
  $\mathbf{d}' := min_{\mathbf{w}}\{\partial_{\mathbf{w}}(Lt_{tau}(G))\}$;
  $L :=$ generates the initial pairs from G whose $\mathbf{w}$ degree is $\mathbf{w}$ lesser than $\mathbf{d}'$;
  $\mathbf{d} := \partial_{\mathbf{w}}(\mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w}} - \langle (G)\rangle^{\mathbf{w}}))$;
  while $L \neq \emptyset$ and $\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) <_{\mathbf{w}} \mathbf{d}'$ **do begin**
    choose $(g_i, g_j) \in L$ s.t. $\mathbf{d} = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
    $L{:=}L - \{(g_i, g_j)\}$;
    $f := \mathrm{NF}_{\ \tau}(\mathrm{SPoly}\,(g_i, g_j), F)$;
    if $f \neq 0$ **then** $F'{:=}F' \cup \{h\}$;
    if $\mathbf{d} \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
      begin
        $G'{:=}Interreduce(G')$;
        L,G:=$\mathtt{Upgrade}^{\mathbf{w},\mathbf{d}}(L, G', G)$;
        $L'{:=}\emptyset$;
        $\mathbf{d} := \mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w}} - \langle (G)\rangle^{\mathbf{w}})$;
      endif;
  endwhile;
  return $G \cap F$;
end.

Remark 3.4 *For non-homogeneous module, proposition 3.3 doesn't hold. To compute a minimal basis for a non-homogeneous module, we have to compute the syzygy module of M, and use this information to delete the redundant elements among the generators.*

3.2 Syzygies

Definition 3.5 *Let $R := k[x_1, \dots, x_n]$, $N \subset R^m$ be a R module and $G := \{g_1, \dots, g_r\}$ a Gröbner basis of N. Let the module morphism ϕ be defined by*

$$\begin{array}{cccc} \phi: & R^r & @>>> & R^m \\ & \mathbf{e}_i & \rightsquigarrow & g_i \end{array}$$

Then the R module $Syz(G) := \mathrm{Ker}\,(\phi) \subset R^r$ it the syzygy module *of G.*

Proposition 3.6 *Let R, N, G and ϕ be as in the definition 3.5. Let $\mathrm{Lt}(g_i) := t_i\mathbf{e}_{n_i}$, ψ be the map defined by*

$$\begin{array}{cccc} \psi: & R^r & @>>> & R^m \\ & \mathbf{e}_i & \rightsquigarrow & t_i\mathbf{e}_{n_i} \end{array}$$

and $S_{ij} := t_j/GCD(t_i, t_j)$, where (i,j) are the pairs of the Gröbner basis G. Then the syzygy module of G is generated by the so-called Taylor basis $\Gamma := \{S_{ij}f_i - S_{ji}f_j\}$.

Remark 3.7

1. *A subset of Γ is sufficient to generate K; practical ways to determine such a subset are given in [Bu] and [GM]. The "useless" elements of Γ, e. g. the elements outside of the sufficient subset, correspond to the useless pairs in the Buchberger algorithm.*

2. *The Buchberger algorithm applied to G (as described in [Bu] and [GM]) checks that for every pair $(g_j, g_j) \in G$, $\text{SPoly}\,(g_i, g_j) \to 0$, e. g. we have $\text{SPoly}\,(g_i, g_j) = \sum c_r S_r g_{i_r}$, for suitable constants c_r and terms s_r, and the sequence $s_r t_{i_r}$ is strictly decreasing. Then $g = g_{i,j} - \sum c_r s_r f_{i_r}$ is a relation between the g_l. It is proved in [MM] that the polynomial g generate the module of relations.*

3. *In the ideal case, another criterion for useless pairs is known, namely a pair (g_i, g_j) is useless if $\text{Lt}(g_i)$, $Lt(g_j)$ are coprime; however these pairs are necessary while computing the syzygies, even if their corresponding syzygy is simple, consisting of $g_i g_j - g_j g_i$=0.*

4. *If $G := \{g_1, \ldots, g_r\}$ is not a Gröbner basis, we can find a (redundant) Gröbner basis $G' \supseteq G$ with the Buchberger algorithm. As a subproduct of the Buchberger algorithm we find the relations between the G': if a pair simplifies to 0, then it gives a relation as above; if it does not, then adding the result of the simplification to G' one restores the fact that pair simplifies to 0.*

From a set of generators of the relations between the G', one recovers a set of generators of the relations between the original G by back substitution: whenever we add an element to G', we add to the relations one stating that the new element is linear combination of the preceding ones, so relations between the G are obtained easily. The idea is very similar to obtaining the Bézout identity from the euclidean algorithm; hence, a better algorithm can mimick the extended euclidean algorithm. Namely, instead of considering elements of M, we consider pairs consisting of an element of M and a vector; we start with the elements (g_i, v_i).

Whenever in the Buchberger algorithm we perform a computation on the first component, we perform the same operation on the second component. In this way, any pair (g, v) that we compute satisfies the assertion $g = \langle v, G\rangle$, where $\langle \_, \_\rangle$ denotes the canonical extension of the scalar product. Whenever in the Buchberger algorithm we obtain a zero (as first element of the pair) we have obtained a relation between the G_i. One sees easily that the result is precisely the same that one would have obtained with the procedure sketched above. We remark now that this procedure (we can call it "extended Buchberger", copying from the extended Euclid algorithm), can be implemented as a Buchberger algorithm for modules, embedding everything in a suitable polynomial ring, where the computations then take place. Let us examine this process in detail.

Proposition 3.8 *Let $R := k[x_1, \ldots, x_n]$, N be an R module generated by $F := \{f_1, \ldots, f_r\}$ and $\mu \in \text{MTO}(N)$. With the notation of theorem 1.88, let $\mathbf{y}$ be new variables, $\mathbf{E}$ be new modules variables and $m_i \in k[\mathbf{x}, \mathbf{E}, \mathbf{y}]$ be terms linearly independent over $k[\mathbf{x}, \mathbf{E}]$. Let $\phi : N \to k[\mathbf{x}, \mathbf{E}]$ be a module morphism fulfilling the hypothesis of theorem 1.88. Consider the module morphism*

$$\begin{array}{cccc} \chi : & N & \to & k[\mathbf{x}, \mathbf{E}, \mathbf{y}] \\ & f_i & \rightsquigarrow & \phi(f_i) + m_i \ , \end{array}$$

and put on $k[\mathbf{x}, \mathbf{E}, \mathbf{y}]$ an ordering τ'' compatible with χ, τ' and such that the variables $\mathbf{y}$ are lexicographically lesser than the others. Compute with the module Buchberger algorithm the Gröbner basis of $\chi(N)$ with respect to τ''. Then the resulting elements belonging to $(\mathbf{y})k[\mathbf{x}, \mathbf{E}, \mathbf{y}]$ are a Gröbner basis of the syzygy module with respect to τ', while the $\mathbf{y}$-homogeneous components of degree 0 of the remaining elements are a Gröbner basis of N with respect to τ'.

Proof. Follows from the embedding definition and the discussion above. □

Remark 3.9 *If we discard the critical pairs between elements in $(\mathbf{y})k[\mathbf{x}, \mathbf{E}, \mathbf{y}]$, we don't compute a Gröbner basis of the syzygy module, but only an (interreduced) system of generators. It is easy to find examples where computing first the syzygies and then their Gröbner bases is advantageous, but it is equally easy to find examples where it is not.*

Example 3.10 *Let R, M, ϕ and τ be as in the example 1.100. The terms $y_1, y_1^2, y_1^3 \in k[x,y,z,E,y_1]$ are linearly independent over $k[x,y,z,E]$.*

We use as embedding the morphism

$$\begin{array}{rcl} \chi: \; M & \to & k[x,y,z,E,y_1] \\ f_1 & \rightsquigarrow & \phi(f_1) + y_1 = x^2E + xyE^2 - y^2E - yzE^2 + y_1 \\ f_2 & \rightsquigarrow & \phi(f_2) + y_1^2 = x^3E - z^3E + xyE^2 - z^2E^2 + y_1^2 \\ f_3 & \rightsquigarrow & \phi(f_3) + y_1^3 = xyE^2 + zE^2 + y_1^3 \end{array}$$

and put on $k[x,y,z,E,y_1]$ the term-ordering τ' associated to the matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

χ and τ' fulfill our desiderata.

The Gröbner basis of $\chi(M)$ in $k[x,y,z,E,y_1,y_1]$ with respect to the ordering τ' is

$x^2E - y^2E - yzE^2 - zE^2 - y_1^3 + y_1$,

$xyE^2 + zE^2 + y_1^3$,

$xy^2E - z^3E + xzE^2 - 2z^2E^2 - zE^2 + xy_1^3 - xy_1 - zy_1^3 - y_1^3 + y_1^2$,

$y^4E + y^3zE^2 - xz^3E + x^2zE^2 + y^2zE^2 - 2xz^2E^2 - xzE^2 + x^2y_1^3 - x^2y_1 + y^2y_1^3 - y^2y_1 - xzy_1^3 - xy_1^3 + xy_1^2$,

$yz^4E^2 - x^3zE^2 + 2x^2z^2E^2 - y^2z^2E^2 + z^4E^2 + x^2zE^2 - y^2zE^2 - x^3y_1^3 + x^3y_1 + x^2zy_1^3 + z^3y_1^3 - z^3y_1 + x^2y_1^3 - x^2y_1^2 - y^2y_1^3 + y^2y_1^2$,

$x^4zE^2 - 2x^3z^2E^2 - xz^4E^2 + z^5E^2 - x^3zE^2 - yz^3E^2 - yz^2E^2 + x^4y_1^3 - x^4y_1 - x^3zy_1^3 - xz^3y_1^3 + xz^3y_1 + z^4y_1^3 - x^3y_1^3 + x^3y_1^2 + xy^2y_1^3 - xy^2y_1^2 - yz^2y_1^3 - yzy_1^3$,

$x^4yy_1^3 - x^4yy_1 - x^3yzy_1^3 - xyz^3y_1^3 + xyz^3y_1 + yz^4y_1^3 - x^3yy_1^3 + x^3yy_1^2 + xy^3y_1^3 - xy^3y_1^2 - x^3zy_1 + x^2z^2y_1^3 - y^2z^2y_1^3 + z^4y_1 + x^2zy_1^2 - y^2zy_1^2$.

The last element is the only one belonging to $(y_1)k[x,y,z,E]$, thus the syzygy module of M is generated by its counterimage $[x^4y - xyz^3 + x^3z - z^4, -x^3y + xy^3 - x^2z + y^2z, -x^4y + x^3yz + xyz^3 - yz^4 + x^3y - xy^3 - x^2z^2 + y^2z^2]$.

Remark that the y_1-homogeneous component of degree 0 of the others element are the same Gröbner basis of N previously computed in the example 2.22.

For another approach that computes at the same time the Gröbner basis and its syzygies, see [MMT]

3.3 Resolutions

Definition 3.11 *Let A be a ring, M, $\{M_i \neq \emptyset\}_{i \in S \subset \mathbb{N}}$ free A modules and*

$$\mathcal{C} : \cdots @ >>> M_i @ >>> M_{i-1} @ >>> M_1 @ >>> M_0 = M \quad i \in S$$

be a long exact sequence of A modules. Then $\mathcal{C}$ is a free resolution *of M. If M is graded and the morphisms of $\mathcal{C}$ are graded morphism of degree 0, then $\mathcal{C}$ is a* graded free resolution *of M. The number $\#S$ is the* length *of $\mathcal{C}$. Let $\mathcal{C}$ be a resolution of M. If there exists no other resolution of M whose length is lesser than $\ell(M)$, Then $\mathcal{C}$ is a* minimal *resolution of M*

Theorem 3.12 (Hilbert syzygy theorem) *Let $R; = k[x_1, \dots, x_n]$ and M an finitely generated graded R module. Then M has a finite graded free resolution of length $\leq n$, by finitely generated free R modules*

Theorem 3.13 *Let M be a graded A module. Then every graded free minimal resolution of M has the same length. Moreover, the ranks and the shifts of the modules in the resolution are the same.*

Proof. It follows easily from the Hilbert function theory of section 1.4. □

Remark 3.14 *A simple way to build a free resolution of a module M is to compute M_1 a syzygy module of M, M_2 a syzygy module of M_1 and so on. Then*

$$\mathcal{C} \cdots @>>> M_i @>>> M_{i-1} @>>> M_1 @>>> M_0 = M$$

is a free resolution of M.

3.4 A Buchberger Algorithm for resolutions

The idea is to embed the data in a suitable polynomial ring and execute here a modified version of the homogeneous Buchberger algorithm for modules. A modification of the tail addition technique introduced in the tag variable syzygy algorithm is used to compute directly the minimal syzygies of every order. To keep the notation and the algorithm simple, the graduation of supposed to have type one. The higher type case is discussed in subsection 3.4.4

3.4.1 The environment

It is possible, using theorem 1.88, to embed the resolution of a graded R module M in a suitable polynomial ring with respect to any ordering on the different syzygy modules in the resolution. The direct product on the modules, with on it a product module term-ordering is considered, and its ordered embedding in a polynomial ring is done in according with theorem 1.88. Here the interest lies in minimal graded resolutions, hence just two term-orderings are examined in detail, the "minimal" term-ordering preserving the degrees, and the Shreyer's ordering, ref. 3.4.5, (see [Sh]).

Proposition 3.15 *Let $R := k[x_1, \dots, x_n]$, M be a $\mathbf{w}$ graded R-module, where $\mathbf{w} := ((a_1, \dots, a_n))$ and*

$$\begin{array}{cccccccccc} & & & M_k & & & & M_0 & & M_{-1} \\ \mathcal{C}: & 0 & \longrightarrow & R(\alpha_{k,1}) \oplus \ldots \oplus R(\alpha_{k,l_k}) & \xrightarrow{\psi_k} & \ldots & \xrightarrow{\psi_1} & R(\alpha_{0,1}) \oplus \ldots \oplus R(\alpha_{0,l_0}) & \xrightarrow{\psi_0} & M \end{array}$$

its minimal graded free resolution. Let μ be the term-ordering on $k[\mathbf{x}, S, D, T]$ defined by the matrix

$$\left[\begin{array}{ccc|ccc} a_1 & \cdots & a_n & 0 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 0 & 0 \\ & & & 0 & 0 & 0 \\ & \left[\ \Gamma_\tau\ \right] & & \vdots & \vdots & \vdots \\ & & & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{array}\right]$$

where Γ_τ is the matrix associated to the term-ordering τ on $k[\mathbf{x}]$. Let S, T be new module variables and D be a new variable. Define $\boldsymbol{\alpha}_{-1}$ as $\mathbf{0}$. Then the module morphism

$$\begin{array}{cccc} \phi: & \displaystyle\bigoplus_{i=-1}^{k} \bigoplus_{j=1}^{l_k} R_i(\alpha_{i,j}) & \longrightarrow & k[\mathbf{x}, S, D, T] \\ & \mathbf{e}_{i,j} & \rightsquigarrow & S^{i+1} D^{-\alpha_{i,j}} T^j \end{array}$$

is injective and ϕ is homogeneous of degree 0 if the weights on $k[\mathbf{x}, S, D, T]$ are $(a_1, \dots, a_n, 0, 1, 0)$.

Proof. Easy check. □

Remark 3.16 *The computations during the Buchberger algorithm for resolutions are performed on* $(k[\mathbf{x}, S, D, T], \mu)$. *The ordering* μ *is not a ring term-ordering, but its restriction to* $\operatorname{Im} \phi$ *is.*

Remark 3.17 *The embedding* ϕ *keeps track of the degrees (shifts) using the variable* D. *The two module variables* S *and* T, *are used as position markers, within the resolution and the modules respectively.*

Remark 3.18

- *The term-ordering on* $\bigoplus_{i=-1}^{k} \bigoplus_{j=1}^{l_k} R_i(\alpha_{i,j})$ *induced by* ϕ, μ *is compatible with the shifts.*
- *The term-ordering induced by* μ *on the image of* M_0 *is* τ. *The orderings induced on the images of* M_i *are degree-compatible term-orderings. The same holds for the induced term-orderings on* M_i.
- *The variables* S *and* T *are lexicographically lesser than the others.*
- *Between terms of a given weighted degree, the greater is the one with the lesser exponent of* S. *I. e. the components of the resolution are arranged by* σ *in decreasing order.*

Definition 3.19 *Let* $f \in k[\mathbf{x}, S, D, T]$. *Then*

- S*-head* (f) *is the homogeneous component of* f *of lesser* S*-degree.*
- S*-tail* $(f) = f - S$*-head* (f).
- S*-ecart* $(f) = Degree_S(S\text{-}head\,(f)) - Degree_S(S\text{-}tail\,(f))$.

Example 3.20 *Let*

$$M := ([0, t^2, x^2 - t^2], [0, 0, y^3 - t^3], [0, 0, z^4 - t^4]) \subset (k[\mathbf{x}])^3$$

be a module over the ring $R := k[x, y, z, t]$ *graded with the standard graduation. A minimal graded free resolution of* M *is*

M_1 $\qquad\qquad$ M_0

$0 \quad @>>> \quad R(-7) \quad @>\psi_1> \qquad > \quad R(-2) \oplus R(-3) \oplus R(-4) \quad @>\psi_0>$

$\mathbf{e}_{11} \rightsquigarrow [0, -t^4 + z^4, t^3 - y^3]$

$$\begin{array}{lcl} \mathbf{e}_{01} & \rightsquigarrow & [0, t^2, x^2 - t^2] \\ \mathbf{e}_{02} & \rightsquigarrow & [0, 0, y^3 - t^3] \\ \mathbf{e}_{03} & \rightsquigarrow & [0, 0, z^4 - t^4]. \end{array}$$

On $k[x, y, z, t, S, D, T]$ *there is the term-ordering* μ *defined by the matrix*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

the embedding is

$$\phi: \quad M_1 \oplus M_0 \oplus M_{-1} \quad @> \qquad >> \quad k[x,y,z,t,S,D,T]$$

$$\begin{array}{lcl}
\mathbf{e}_{11} & \rightsquigarrow & S^2D^7T \\
\mathbf{e}_{01} & \rightsquigarrow & SD^2T \\
\mathbf{e}_{02} & \rightsquigarrow & SD^3T^2 \\
\mathbf{e}_{03} & \rightsquigarrow & SD^4T^3 \\
\mathbf{e}_{-1\,1} & \rightsquigarrow & T \\
\mathbf{e}_{-1\,2} & \rightsquigarrow & T^2 \\
\mathbf{e}_{-1\,2} & \rightsquigarrow & T^3
\end{array}$$

Hence, the generators of M is interpreted through ϕ as

$$\begin{array}{lcl}
[0,t^2,x^2-t^2] & \rightsquigarrow & t^2T^2+x^2T^3-t^2T^3 \\
[0,0,y^3-t^3] & \rightsquigarrow & y^3T^3-t^3T^3 \\
[0,0,z^4-t^4] & \rightsquigarrow & z^4T^3-t^4T^3
\end{array}$$

and the generator of M_0 as

$$[0,-t^4+z^4,t^3-y^3] \rightsquigarrow -t^4SD^3T^2+z^4SD^3T^2+t^3SD^4T^3-y^3SD^4T^3$$

If $f = xyD^3T^2 + x^2yD^2T + xSD^4T + x^3SD^2T$, then

- *S-ecart $(f) = 1$;*
- *S-head $(f) = xyD^3T^2 + x^2yD^2T$;*
- *S-tail $(f) = xSD^4T + x^3SD^2T$.*

3.4.2 The algorithm

The idea, borrowed from the tag variable syzygy algorithm, is to keep track of the computations on an element through the addition of a term in additional variables "tail". To avoid special cases, and additional notation, the input polynomials are considered interreduced. The operations are done in the module environment.

The pairs are processed and kept in according with the ordering described in proposition 3.15, using the same method as in the minimal basis algorithm, 3.1, since the problems are quite similar. At start, the Gröbner basis C is empty, and in the pair list L there are the input polynomials, as singleton pairs. The S-polynomial of a singleton pair is the polynomial itself.

The pairs in L of least degree d are then reduced by C, and the results are kept in the auxiliary list C'. During the addition to C', to the resulting polynomials is added a tail if necessary 3.4.2.The reduction procedure is described just below in 3.4.2. When all the pairs of degree d have been processed, the polynomials in C' are interreduced, to the resulting polynomials is added a tail if necessary 3.4.2, and the pairs of the resulting polynomials are added to L and the polynomial themselves are added to C. The list C' is emptied.

As noted in the discussion of the minimal basis algorithm 3.1, at the moment of the insertion in the basis C the polynomials are in definitive form: no further insertion will ever be able to modify them.

Hence, the algorithm described above is a mix of the minimal basis algorithm, and the tag variable syzygy algorithm.

Simplification

The lists $C_i = \{c_{i,1}, \ldots, c_{i,k_i}\}$ collect the elements of C of degree i. The lists C_i^* collect the marked elements of C_i.

The variable S is a module variables. Let χ be the current polynomial processed by the Buchberger algorithm. The tail addition strategy will assure that it will be of S-ecart 0 or 1. The simplifications allowed are different in the two cases:

- if χ is of S-ecart 1 and the monomial being simplified is in the S-head, then it is simplified with elements of C.
- Otherwise χ is simplified with elements of $\{S\text{-head}\,(f) \mid f \in C\}$

Definition 3.21 *The procedure described above is the Special Normal Form procedure* (SNF).

Tail addition

When $\chi := a\,\mathbf{x}^{\boldsymbol{\alpha}} S^i D^k T^l + \cdots$ is no longer simplifiable, if χ is of S-ecart 0 then S-tail $(\chi) := S^{i+1} D^{|\alpha|+k} T^j$, where j is $\#C_i + \#C_i^{'}$. The element $\chi + S$-tail (χ) is marked and will referred to as $c^*_{i,j}$. Let $a_{i,j}$ be $|\boldsymbol{\alpha}| + k$.

Remark 3.22

a) The procedure SNF *terminates since the usual procedure Normal Form terminates.*

b) If the S-ecart of χ is at most 1 it will remain so throughout and at the end of the reduction. item[g)] The points a) through d) of remark 2.12 are valid for the procedure SNF

Remark 3.23 *As is shown in the subsequent subsection, and similarly to the tag variable syzygy algorithm, if S-ecart $(\chi) = 0$, the element χ gives a syzygy, hence a new generator is added to the opportune syzygy module in the resolution. The minimality of this syzygy is proved in theorem 3.26. The position of this generator in the resolution is given by the exponent of S, its position in the syzygy module in given by the exponent of T and the corresponding shift is given by its weighted degree. Moreover, the tail and reduction strategies assure that whenever in a term S, T appears with exponents i, j, then in the same term D will appear with exponent $a_{i,j}$;*

Throughout the algorithm, $\mathbf{w}$ is the weighted degree defined in proposition 3.15.

Buchberger Algorithm for Resolutions

Input $G := \{g_1, \ldots, g_r\} \subseteq k[\mathbf{x}, S, D, T]$, g_i homogeneous;
Output $\{R_1, \ldots, R_k\}$, where R_i is a minimal basis of the i^{th} syzygy module of (G) and k is the length of a minimal graded free resolution of (G).

begin
  $C := \emptyset$; $C' := \emptyset$;
  L:= generates the initial pairs;
  d:=$min\{wdeg(SPoly(g_i, g_j)) \mid (g_i, g_j) \in L\}$;
  while $L \neq \emptyset$ **do begin**
    choose $(g_i, g_j) \in L$ s.t. $d = wdeg(SPoly(g_i, g_j))$;
    L:=$L - \{(g_i, g_j)\}$;
    f:=$\texttt{SpecialNormalForm}_\sigma(SPoly(g_i, g_j), C)$;
    if $f \neq 0$ **then**
      begin

```
            if  S-ecart (h) = 0 then
               begin
                  h:=h + Tail(h)}; mark h;
                  C':=C' ∪ {h};
               else
               C':=C' ∪ {h};
            endif;
         endif
      if d ≠ min{wdeg(SPoly(g_i, g_j)) | (g_i, g_j) ∈ L} then
         begin
            C':=Interreduce(C');
            L,C:=Upgrade(L, C', C);
            L':=∅;
            d:=min{wdeg(SPoly(g_i, g_j)) | (g_i, g_j) ∈ L};
         endif;
   endwhile;
   return  {f ∈ B | f marked and wdeg(f) = i}_{i∈ℕ};
end.
```

Procedure `SpecialNormalForm`

Input $f \in k[\mathbf{x}, S, D, T]$; $C \subset k[\mathbf{x}, S, D, T]$.
Output the special normal form of f.

```
begin
   h:=0;
   while S-ecart (f) = 1 do
      begin
         while exists ∃g ∈ G s.t. Lt(g) | Lt(f) do
            begin
               choose g ∈ G s.t. t Lt(g) = Lt(f);
               f:=f − t g;
            endwhile;
         if S-ecart (f) = 1 then
         begin
            h:=h + Lt(f);
            f:=f − Lt(f);
         endif;
      endwhile;
   return h+NormalForm(f, {S-head (f) | f ∈ C});
end.
```

Procedure Upgrade

Input L, C', $C \subset k[\mathbf{x}, S, D, T]$.
Output L, C.

```
begin
  B_i:=∅;
  foreach h ∈ L' do
    begin
      if S-ecart (h) = 0 then
        begin
          h:=h + Tail(h)}; mark h;
          C:=C ∪ {h};
        else
          C:=C ∪ {h};B := B ∪ {S-Head(h)};
        endif;
      L:=Upgrade the S-Polys in L with respect to C and h using the G-M criteria
    endfor;
  return L, G;
end.
```

3.4.3 Termination and correctness

Since the resolution algorithm is a modified Buchberger algorithm, termination and correctness of the output have to been proved. The notation used in this subsection have been introduced above.

Proposition 3.24 *In the course of the algorithm, if $f \in C$ then $f = f_i + f_{i+1}$, f_i, f_{i+1} homogeneous in S of degrees i and $i+1$ respectively, and substituting in f_{i+1} the term $\mathbf{x}^{\alpha_n} S^i D^{c_{i,j}} T^j$ with $c_{i,j}$ we obtain f_i. Hence, f_{i+1} is the syzygy of some elements in M_i. Moreover, as above, $f_i \in M_0$ or $f_i \in \operatorname{Syz}(M_{i-1})$.*

Proof. This follows easily from the definition of tail 3.4.2 and of the reduction strategy 3.4.2. □

Remark 3.25 *Let g be the result of the reduction of f with the strategy explained above and the final interreduction of the same-degree polynomials. Then, since S-polynomials other than the initial ones are created from elements of C (S-ecart$= 1$), for an S-polynomial f, one of the subsequent proposition holds:*

1) *the S-ecart of f is 0, (f is the special S-polynomial of an element of the input) and then f is reduced by the S-heads of elements in C only. The resulting polynomial belongs to the Gröbner basis of M_0.*

2) *the S-ecart of f is 1, hence $f \in M_i \oplus M_{i+1}$. By proposition 3.24, S-head$(f) \in \operatorname{Syz}(M_{i-1}) = M_i$ and S-tail$(f) \in \operatorname{Syz}(M_i) = M_{i+1}$. Then while S-ecart$(f) = 1$, f is reduced by elements in C, whose S-heads and S-tails belong to M_i and M_{i+1} respectively, and then by the S-heads of elements in C, which belong to M_{i+1}. Hence,*

   a) *If $g = 0$, then the element f is the Gröbner combination of elements in M_i, and its syzygy is the Gröbner combination of elements in M_{i+1}.*

   b) *If $g \neq 0$ and S-ecart$(g) = 0$, then S-head$(f) \to 0$ and a syzygy of M_i has been found, S-tail(f), which is then an element of M_{i+1}; a tail is added to it, the element is marked and it is added to C.*

   c) *If $g \neq 0$ and S-ecart$(g) \neq 0$, then (S-head(g)) is an element of the Gröbner basis of M_i, with its tail (S-tail(g)) $\in M_{i+1}$. Then g is added to C.*

Theorem 3.26 *Let the polynomials in* $\cup_{j \le d} C_j$ *of* S*-degree* i *be* $C_{d,i}$*. Using the notation introduced above, after the computation of the pairs in degree* d*, for each* $i \ge 0$

i) the elements belonging to $C_{d,i}$ *are the reduced Gröbner basis of* M_i *truncated to degree* d *with respect to the term-ordering described in remark 3.18;*

ii) the elements belonging to $C_{d,i}$ *are a minimal basis of* M_i

Proof.

i) From remark 3.25, and the description of the algorithm, it follows that:

> the algorithm computes the reduced Gröbner basis of M_0 truncated in degree d with respect to the term-ordering on M_0.
>
> For the syzygy module M_i, the generators in degree $\le d$ are generated by the computation of the Gröbner basis of the syzygy module M_{i-1}.
>
> The resolution algorithm computes the reduced Gröbner basis of M_i with respect to the degree-compatible term-ordering described by corollary 3.18.

ii) Form i), the elements of $C_{d,i}$ are a Gröbner basis of (M_i). The unmarked elements of $C_{d,i}$ are reductions of S-polynomials of the marked elements, hence their polynomial combination, hence are not part of the minimal basis. Thus, the marked elements are a basis by themselves, and, since they have survived the reduction process, a minimal basis. □

Remark 3.27 *No element which is not in some minimal or Gröbner basis is ever created. Hence, it is not necessary to remove any element already found by checking some syzygy, and no syzygy refers to such an element. In this sense, the algorithm is minimal.*

Hilbert Syzygiensatz guarantees that a finite minimal resolution of M exists. Hence, after a finite number of steps, the algorithm has found one such resolution by theorem 3.26. Gröbner bases for every syzygy module has been computed, hence the list L is empty, and the algorithm stops. The shifts of the resolution are given by the degree of the marked elements in the lists B_i (ref. remark 3.23). Hence,

Corollary 3.28 *Let* M *be an homogeneous* R *module. Them the Buchberger algorithm for resolution, with as input the generators of* M*, terminates and computes a minimal graded resolution of* M*.*

By modifying the tail addition procedure this algorithm computes

- the Gröbner basis of the input, if no tail is added;
- the syzygies of the input, if tail addition is performed only for elements of S-degree 0;
- the minimal resolution of the input, with the full tail addition.

No modifications are necessary to switch from ring to module resolution computations, apart from the trivial ones.

3.4.4 Multigraded case

If the R module M is graded with respect to the system of weights $\mathbf{w}_1, \dots, \mathbf{w}_r$, of type > 1, some modifications are necessary to exploit the multigraded structure :

- A multihomogeneous version of the Buchberger algorithm has to be used.
- A new module variable D_i is introduced for each weight to keep track of the degree with respect to that weight.
- The term-ordering σ on $k[\mathbf{x}, S, \mathbf{D}, T]$ is defined by the matrix

$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & \ddots & 0 & \vdots \\ w_{r,1} & \cdots & w_{r,n} & 0 & 0 & 0 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 \\ \multicolumn{3}{c}{\multirow{4}{*}{$\begin{bmatrix} & & \\ & \Delta_\tau & \\ & & \end{bmatrix}$}} & 0 & 0 & \cdots & 0 & 0 \\ & & & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

  where $w_1, \dots, w_n$ are the weights of $x_1, \dots, x_n$ and Δ_τ is the matrix generated by the first $n-r+1$ rows of the matrix associated to the term-ordering τ on $k[\mathbf{x}]$ linearly independent from $\mathbf{w}_1, \dots, \mathbf{w}_r$.

Since the notation in the multigraded case are quite heavy, while the modifications are minor, the details are left to the reader.

3.4.5 Other orderings

The term-ordering chosen in the Buchberger algorithm for resolutions keeps track of the weighted degree of a pair with minimal overhead (the introduction of one variable). Any other term-ordering on $k[\mathbf{x}, S, D, T]$ compatible with ϕ, μ which fulfills the conditions of remark 3.18 may be used in this resolution algorithm. Since the Schreyer ordering is well known for its utility in syzygy and resolution computations [Sh], [CDNR], it is here examined in detail.

Definition 3.29 *Let M be a R module, let*

$$\mathcal{C} : 0 \longrightarrow M_k \xrightarrow{\psi_k} \dots \xrightarrow{\psi_1} M_0 \xrightarrow{\psi_0} M = M_{-1}$$

be a minimal resolution of M and $\mathbf{e}_{i,j}$ the j^{th}-generator of the i^{th}-module. Let $\mu_k, \dots, \mu_{-1}$ be module term-orderings on $M_k, \dots, M_{-1}$ and let $Lt_{\mu_i}(\psi_i(\mathbf{e}_{i,j}))$ be $t_{i,j}\mathbf{e}_{i,m}$. Let $\mathbf{x^a}\mathbf{e}_{i,j}$ and $\mathbf{x^b}\mathbf{e}_{i,j'}$ be module terms. If $\mu_k, \dots, \mu_{-1}$ are such that

$$\mathbf{x^a}\mathbf{e}_{i,j} >_{\mu_i} \mathbf{x^b}\mathbf{e}_{i,j'} \iff \mathbf{x^a} t_{i,j}\mathbf{e}_{i,m} >_{\mu_{i-1}} \mathbf{x^b} t_{i,j'}\mathbf{e}_{i,m'} \text{ or } \mathbf{x^a} t_{i,j}\mathbf{e}_{i,m} = \mathbf{x^a} t_{i,j'}\mathbf{e}_{i,m'} \text{ and } m < m'$$

then $\mu_k, \dots, \mu_{-1}$ are Schreyer orderings for $\mathcal{C}$.

Let us show the embedding and term-ordering associated to Schreyer orderings for the Buchberger algorithm for resolution.

Proposition 3.30 *With the notation introduced above and in the habitat discussion, let $\mu_k, \dots, \mu_{-1}$ be a Shreyer's ordering for $\mathcal{C}$, a minimal graded resolution of an homogeneous R module M. Let μ_S be the term-ordering associated to the matrix*

$$\begin{bmatrix} w_1 & \cdots & w_n & 0 & 0 \\ 0 & \cdots & 0 & -1 & 0 \\ \multicolumn{3}{c}{\multirow{4}{*}{$\left[\quad \Delta_\tau \quad\right]$}} & 0 & 0 \\ & & & \vdots & \vdots \\ & & & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

where the matrix Δ_τ is defined by the first $n-1$ rows linearly independent from $(w_1, \dots, w_n)$ of the matrix associated to the term-ordering τ on $k[\mathbf{x}]$. Then the module morphism

$$\begin{array}{rcll} \psi : & \bigoplus_{i=-1}^{k} \bigoplus_{j=1}^{l_k} R(\alpha_{i,j}) & \longrightarrow & (k[\mathbf{x}, S, T], \mu_S) \\ & & & \\ & \mathbf{e}_{i,j} & \rightsquigarrow & Lt_{\mu_i}(\psi_i(\mathbf{e}_{i,j}))S^{i+1}T^j \end{array}$$

is injective and $\psi_{|(M_i,\mu_i)}$ is ordered. Moreover, ψ is also homogeneous of degree 0 if the weights of $k[\mathbf{x}, S, T]$ are $(w_1, \dots, w_n, 0, 0)$. Moreover, the

Proof. Easy check. □

Remark 3.31 *Since the purpose of variable D was to keep track of shifts, and this is now done through a different definition of the embedding, this variable is now useless. The price are denser terms to deal with.*

The only modification to the algorithm in order to use Schreyer's ordering is the tail definition. Using the notation introduced earlier,

- for an element χ, with $\mathrm{Lt}(\chi) = x_1^{a_1} \cdots x_n^{a_n} S^i T^j$, we define $Tail(\chi)$ as $x_1^{a_1} \cdots x_n^{a_n} S^{i+1} T^{j'}$.

It is immediate to check that this tail and order definition fulfill the desiderata, and that they give a slight different version of the algorithm described above.

Remark 3.32 *It is always possible to apply Hilbert Driven techniques to the minimal resolution algorithms. If more information is known, like the regularity of the minimal resolution, further optimizations are possible. The savings are often substantial. For a full analysis see[CDNR]. The methods there describes are applicable to our minimal resolution algorithm.*

3.5 An example

Example 3.33 *Let $I := (f_1 := x + y, f_2 := y^2, f_3 := yz^2 + u^3)$ be an homogeneous (with respect to the standard weights) ideal of $k[x, y, z, u]$. The minimal graded resolution of I is*

$$0@>>>R(-6)@>>>R(-3) \oplus R(-4) \oplus R(-5)@>>>R(-1) \oplus R(-2) \oplus R(-3)$$

The generators of I are interreduced; the initial pairs are : $(f_1, 0)_1$(degree1), $(f_2, 0)_2$, $(f_3, 0)_3$.

$(f_1, 0)_1@>>>x + y$. *A minimal element.*

$(f_2, 0)_2 @ >>> y^2$. *A minimal element.*

$C := \{c^*_{1,1} := x + y + SDT, c^*_{2,1} := y^2 + SD^2T^2\}$.

Computation in degree 3

$Pairs := \{(c^*_{1,1}, c^*_{2,1},)_3, (f_3, 0)_3, \}$.

$SPoly(c^*_{1,1}, c^*_{2,1}) = y^3 - SD^2T^2x + SDTy^2 @ > c^*_{2,1} >> -xSD^2T^2 + y^2SDT - ySD^2T^2$. *A minimal element (Its S-degree is 0).*

$SPoly(f_3, 0) = yz^2 + u^3$. *A minimal element.*

$C := C \cup C_3\{b^*_{3,1} := -xSD^2T^2 + y^2SDT - ySD^2T^2 + S^2D^3T, c^*_{3,2} := yz^2 + u^3 + SD^3T^3\}$.

Computations in degree 4

The pairs $\{(c^*_{1,1}, c^*_{3,1}), (c^*_{2,1}, c^*_{3,1}), (c^*_{3,1}, c^*_{3,2})\}$ *are not possible modules pairs, since the exponents of* S, T *of their leading terms are different.*

$Pairs := \{(c^*_{1,1}, c^*_{3,2},)_4, (c^*_{2,1}, c^*_{3,2})_4, \}$.

$SPoly(c^*_{1,1}, c^*_{3,2}) = y^2z^2 - xu^3 - xSD^3T^3 + yz^2SDT @ > c^*_{2,1} >> -xu^3 - xSD^3T^3 - z^2SD^2T^2 + yz^2SDT @ > c^*_{1,1} >> yu^3 - xSD^3T^3 - z^2SD^2T^2 + yz^2SDT + u^3SDT =: g_1$.

$SPoly(c^*_{2,1}, c^*_{3,2}) = -yu^3 - ySD^3T^3 + z^2SD^2T^2 =: g_2$.

$g_1 @ > g_2 >> -xSD^3T^3 - ySD^3T^3 + yz^2SDT + u^3SDT$. *A minimal element.*

$C := C \cup C_4 := \{c^*_{4,1} := -xSD^3T^3 - ySD^3T^3 + yz^2SDT + u^3SDT + S^2D^4T^2, c_{4,2} := -yu^3 - ySD^3T^3 + z^2SD^2T^2\}$.

Computations in degree 5

The pairs $(c^*_{1,1}, c^*_{4,1}), (c^*_{2,1}, c^*_{4,1}), (c^*_{3,1}, c^*_{4,1}), (c^*_{3,2}, c^*_{4,1})(c^*_{3,1}, 6c_)), (c^*_{4,1}, c_{4,2})$ *are useless.*

$Pairs := \{(c^*_{1,1}, c_{4,2},)_5, (c^*_{2,1}, c_{4,2})_5, (c^*_{3,2}, c_{4,2})_6\}$.

$SPoly(c^*_{1,1}, c_{4,2}) = y^2u^3 - SD^3T^3xy + SD^2T^2xz^2 + SDTyu^3 @ > c^*_{2,1} >> -xySD^3T^3 + xz^2SD^2T^2 - u^3SD^2T^2 + yu^3SDT @ > b^*_{4,1} >> y^2SD^3T^3 + xz^2SD^2T^2 - y^2z^2SDT - u^3SD^2T^2 @ > b^*_{3,1} >> y^2SD^3T^3 - yz^2SD^2T^2 - u^3SD^2T^2 =: g_1$. *A minimal element.*

$SPoly(c^*_{2,1}, c_{4,2}) = -y^2SD^3T^3 + yz^2SD^2T^2 + u^3SD^2T^2 =: g_2$.

$g_1 @ > g_2 >> 0$.

$B := B \cup \{b^*_{6,1} := -y^2SD^3T^3 + yz^2SD^2T^2 + u^3SD^2T^2\}$.

$C := C \cup C_5\{c^*_{5,1} := -y^2SD^3T^3 + yz^2SD^2T^2 + u^3SD^2T^2 + S^2D^5T^3\}$.

Computation in degree 6

The pairs $(c^*_{1,1}, c^*_{5,1}), (c^*_{2,1}, c^*_{5,1}), (c^*_{3,2}, c^*_{5,1}), (c_{4,2}, c^*_{5,1})$ *are useless.*

$Pairs := \{(c^*_{3,2}, c_{4,2})_6, (c^*_{3,1}, c^*_{5,1})_6, (c^*_{4,1}, c^*_{5,1})_6\}.$

$SPoly(c^*_{3,1}, c^*_{5,1}) = y^3SD^3T^3 - y^4SD^2T^2 + xyz^2SD^2T^2 + xu^3SD^2T^2 + xS^2D^5T^3 - y^2S^2D^4T^2@ > c^*_{5,1} >> -y^4SD^2T^2 + xyz^2SD^2T^2 + SD^2T^2y^2z^2 + SD^2T^2xu^3 + SD^2T^2yu^3 + S^2D^5T^3x + S^2D^5T^3y - S^2D^4T^2y^2@ > c^*_{3,1} >> -y^4SD^2T^2 + y^3z^2SDT + xu^3SD^2T^2 + yu^3SD^2T^2 + xS^2D^5T^3 + yS^2D^5T^3 - y^2S^2D^4T^2 + yz^2S^2D^3T@ > c^*_{3,1} >> -y^4SD^2T^2 + y^3z^2SDT + y^2u^3SDT + xS^2D^5T^3 + yS^2D^5T^3y - y^2S^2D^4T^2 + yz^2S^2D^3T + u^3S^2D^3T.$

$SPoly(c^*_{4,1}, c^*_{5,1}) = y^3SD^3T^3 + xyz^2SD^2T^2 - y^3z^2SDT + xu^3SD^2T^2 - y^2u^3SDT + xS^2D^5T^3 - y^2S^2D^4T^2@ > c^*_{5,1} >> xyz^2SD^2T^2 + y^2z^2SD^2T^2 - SDTy^3z^2 + SD^2T^2xu^3 + yu^3SD^2T^2 - y^2u^3SDT + xS^2D^5T^3 + yS^2D^5T^3 - y^2S^2D^4T^2@ > c^*_{3,1} >> xu^3SD^2T^2 + yu^3SD^2T^2 - y^2u^3SDT + xS^2D^5T^3 + yS^2D^5T^3 - y^2S^2D^4T^2 + yz^2S^2D^3T@ > c^*_{3,1} >> xS^2D^5T^3 + yS^2D^5T^3 - y^2S^2D^4T^2 + yz^2S^2D^3T + u^3S^2D^3T$. *A minimal element.*

$SPoly(c^*_{3,2}, c_{4,2}) = u^6 - yz^2SD^3T^3 + z^4SD^2T^2 + u^3SD^3T^3.$

$C := C \cup C_6\{c_{6,1} := -y^4SD^2T^2 + y^3z^2SDT + y^2u^3SDT + xS^2D^5T^3 + yS^2D^5T^3y - y^2S^2D^4T^2 + yz^2S^2D^3T + u^3S^2D^3T, c^*_{6,2} := xS^2D^5T^3 + yS^2D^5T^3 - y^2S^2D^4T^2 + yz^2S^2D^3T + u^3S^2D^3T, c_{6,3} := u^6 - yz^2SD^3T^3 + z^4SD^2T^2 + u^3SD^3T^3\}.$

All pairs of degree ≥ 7 *reduce to zero.*

It is easy to read the Betti numbers from the marked elements in B :

$$0 \longrightarrow \underbrace{\underbrace{R(-6)}_{\substack{c^*_{6,2} \\ 1\ poly\ deg\ 6}}}_{S\text{-}deg\ 2} \longrightarrow \underbrace{\underbrace{R(-3)}_{\substack{c^*_{3,1} \\ 1\ poly\ deg\ 3}} \oplus \underbrace{R(-4)}_{\substack{c^*_{4,1} \\ 1\ poly\ deg\ 4}} \oplus \underbrace{R(-5)}_{\substack{c^*_{5,1} \\ 1\ poly\ deg\ 5}}}_{S\text{-}deg\ 1} \longrightarrow \underbrace{\underbrace{R(-1)}_{\substack{c^*_{1,1} \\ 1\ poly\ deg\ 1}} \oplus \underbrace{R(-2)}_{\substack{c^*_{2,1} \\ 1\ poly\ deg\ 2}} \oplus \underbrace{R(-3)}_{\substack{c^*_{3,2} \\ 1\ poly\ deg\ 3}}}_{S\text{-}deg\ 0}$$

3.6 Statistics

The Buchberger Algorithm for resolutions has complete compatibility with, and make full use of all the usual tricks like Gebauer-Möller criteria etc.

Examples

The examples has been chosen from three classes :

- Determinantal ideals. Few reduction per pair. Polynomials have large supports. Resolutions with few degrees for every level. The schreyer's ordering is the best one.

  $M^2_{3\times 4}$: 2×2 minors of the generic 3×4 matrix.

  $M^2_{4\times 5}$: 2×2 minors of the generic 4×5 matrix.

  $Comm_{3\times 3}$: the ideal of the entries of the matrix $A \cdot B - B \cdot A$, where $A := (x_{i,j})$ and $B := (y_{i,j})$.

- Cyclic ideals. Lots of reductions per pair. Polynomials have large supports. Resolutions levels with several degrees. The minimal ordering is the best one.

C_5 : Homogeneous Cyclic 5.

C_5' : A slight variation of C_5.

C_6 : Homogeneous Cyclic 6, with the homogeneizing variable as the biggest one.

C_7 : Homogeneous Cyclic 7, with the homogeneizing variable as the biggest one. This is the first time, to my knowledge, that this resolution has been computed. The computation with the Shreyer's ordering and the computation with the usual CoCoA resolution algorithm have been interrupted after $20h$ Cpu time.

- Generic polynomial ideals. Lots of reductions per pair, very large supports. Very simple resolutions. The orderings are roughly equivalent.

$D^4_{4,4}$: 4 random dense polynomial of degree 4 in 4 variables.

$D^4_{4,5}$: 4 random dense polynomial of degree 5 in 4 variables.

$D^{5,4}_4$: 4 random dense polynomial of degree 5 in 5 variables.

For every example, are shown two set of statistics, Minimal refers to the computation using the ordering introduced in proposition 3.15, and Schreyer refers to the computation using Schreyer's ordering (subsection 3.4.5). **Pairs** is the number of pairs processed and **Reductions** is the number of reduction performed.

Name	**Pairs**	**Reductions**	**Time**
$M^2_{3\times 4}$ *Shreyer*	277	1260	0.41 s
$M^2_{3\times 4}$ *Minimal*	292	1929	0.71 s
$M^2_{4\times 5}$ *Shreyer*	1183	9439	4.61 s
$M^2_{4\times 5}$ *Minimal*	3591	26539	27.53 s
$Comm_{3\times 3}$ Shr	405	9434	3.36 s
$Comm_{3\times 3}$ Deg	584	17537	12.97 s
$D^4_{4,4}$ Shreyer	196	10653	7.16 s
$D^4_{4,4}$Minimal	196	9163	6.67 s
$D^4_{4,5}$ Shreyer	356	46249	34.84 s
$D^4_{4,5}$ Minimal	343	43226	32.59 s
$D^{5,4}_4$ Shreyer	343	494	640.19 s
$D^{5,4}_4$ Minimal	343	494	529.85 s
C_5 Shreyer	226	7930	1.60 s
C_5 Minimal	191	7917	1.30 s
C_5' Shreyer	329	21383	5.98 s
C_5' Minimal	310	19654	4.81 s
C_6 Shreyer	526	261026	1686.61 s
C_6 Minimal	305	71307	79.28 s
C_7 Minimal	1325	2314971	9 h 340 s

To measure the performance of the resolution algorithm with respect to the CoCoA standard resolution algorithm [CDNR] below there are the comparative timings for the examples.

The statistics for the computations using as coefficient field $\mathbb{Z}_{32003}$.

Name	Minimal	Schreyer	CDNR
$M^2_{3\times 4}$	0.70 s	0.41 s	0.34 s
$M^2_{4\times 5}$	27.53 s	4.61 s	2.71 s
$Comm_{3\times 3}$	12.97 s	3.36 s	1.96
$D^4_{4,4}$	6.67 s	7.16 s	7.54 s
$D^4_{4,5}$	32.59 s	34.84 s	41.33 s
$D^{5,4}_4$	529.85 s	569.06 s	640.19 s
C_5	1.30 s	1.60 s	4.48 s
C'_5	4.81 s	5.98 s	48.61
C_6	79.28 s	1686.61	170.66 s
C_7	9 h 340 s		s

Remark 3.34 *The only example that needs swap memory is C_7, but it needs substantial swap memory (200MB); its timings are hence quite high.*

The statistics for the computations using as coefficient field $\mathbb{Q}$. The examples $D^4_{4,4}$, $D^4_{4,5}$, $D^{5,4}_4$ have not been tested in 0 characteristic since in that case all the time is used up by the numerical computations.

Name	Minimal	Schreyer	CDNR
$M^2_{3\times 4}$	1.42 s	0.81 s	0.67 s
$M^2_{4\times 5}$	43.70 s	8.72 s	5.81 s
$Comm_{3\times 3}$	28.61 s	9.22 s	6.81 s
C_5	8.50 s	11.17 s	46.99 s
C'_5	39.68 s	46.91 s	1195.13 s
C_6	347.76 s		1223.57 s

3.6.1 Conclusions

The algorithm here proposed for the minimal resolution computations is quite efficient in comparison to the classical, syzygy module-by-syzygy module algorithm. With respect to the new algorithms, (e. g. [CDNR]), the results are mixed. There are classes of examples for which our algorithm is the best, and other for which it is not. The difference seems to be the complexity of the graded structure of the resolution, our algorithm gaining when the grading is more complex (e. g. the cyclic examples). See the description of the explicit resolutions in the next page.

Moreover, the choice of ordering is crucial for the computation of a minimal resolution. Which ordering is the best for a given example seems to depend on its class. The data show a certain consistency within classes. The relative behaviour of the algorithms is roughly invariant with respect to the change of the coefficient field from $\mathbb{Z}_{32003}$ to $\mathbb{Q}$. Overall, the performances of the resolution algorithm presented in this paper are roughly comparable to the CoCoA resolution algorithm.

3.7 Examples resolution

Here there are the resolutions of the examples examined in the section 3.6. Except for the dense polynomial examples, the resolution are shown as matrices; The entry i, j of the matrix represents the number of generators of the $n-j$ syzygy module in degree i, where n is the matrix number of rows. For the dense polynomial examples, due to the simplicity of the resolutions, the resolutions are shown as usual. Note that the syzygy modules in the matrix are written in the same order as in the resolution, so the last column represent the generators of the 0 syzygy module.

$M^2_{3\times 4}$

$$\begin{bmatrix} & 3 & 12 & 34 & 60 & 52 & 18 \\ 1 & . & . & . & . & . & . \\ 2 & . & . & . & . & . & 18 \\ 3 & . & . & . & . & 52 & \\ 4 & . & . & . & 60 & & . \\ 5 & . & . & 24 & & . & . \\ 6 & . & . & 10 & . & . & . \\ 7 & . & 12 & . & . & . & . \\ 8 & 3 & . & & . & . & . \end{bmatrix}$$

$M^2_{4\times 5}$

$$\begin{bmatrix} & 6 & 20 & 65 & 132 & 120 & 40 \\ 1 & . & . & . & . & . & . \\ 2 & . & . & . & . & . & . \\ 3 & . & . & . & . & . & 40 \\ 4 & . & . & . & . & 120 & . \\ 5 & . & . & . & 132 & . & . \\ 6 & . & . & 50 & . & . & . \\ 7 & . & . & . & . & . & . \\ 8 & . & . & 15 & . & . & . \\ 9 & . & 20 & . & . & . & . \\ 10 & 6 & . & . & . & . & . \end{bmatrix}$$

$Comm_{3\times 3}$

$$\begin{bmatrix} & 5 & 32 & 61 & 60 & 33 & 8 \\ 1 & . & . & . & . & . & . \\ 2 & . & . & . & . & . & 8 \\ 3 & . & . & . & . & 2 & . \\ 4 & . & . & . & . & 31 & . \\ 5 & . & . & . & 32 & . & . \\ 6 & . & . & 3 & 28 & . & . \\ 7 & . & . & 58 & . & . & . \\ 8 & . & 32 & . & . & . & . \\ 9 & 4 & . & . & . & . & . \\ 10 & 1 & . & . & . & . & . \end{bmatrix}$$

C_5

$$\begin{bmatrix} & 1 & 5 & 10 & 10 & 5 \\ 1 & . & . & . & . & 1 \\ 2 & . & . & . & 1 & 1 \\ 3 & . & . & . & 1 & 1 \\ 4 & . & . & 1 & 2 & 1 \\ 5 & . & . & 1 & 2 & 1 \\ 6 & . & . & 2 & 2 & . \\ 7 & . & 1 & 2 & 1 & . \\ 8 & . & 1 & 2 & 1 & . \\ 9 & . & 1 & 1 & . & . \\ 10 & . & 1 & 1 & . & . \\ 11 & 1 & 1 & . & . & . \end{bmatrix}$$

AC_5

$$\begin{bmatrix} & 1 & 5 & 10 & 10 & 5 \\ 1 & . & . & . & . & 1 \\ 2 & . & . & . & . & 1 \\ 3 & . & . & . & . & 1 \\ 4 & . & . & . & 2 & 1 \\ 5 & . & . & . & 1 & 1 \\ 6 & . & . & . & 2 & . \\ 7 & . & . & 1 & 2 & . \\ 8 & . & . & 2 & 2 & . \\ 9 & . & . & 2 & 1 & . \\ 10 & . & . & 2 & . & . \\ 11 & . & 1 & 1 & . & . \\ 12 & . & 1 & 2 & . & . \\ 13 & . & 2 & . & . & . \\ 14 & . & . & . & . & . \\ 15 & . & 1 & . & . & . \\ 16 & 1 & . & . & . & . \end{bmatrix}$$

C_6

$$\begin{bmatrix} & 2 & 12 & 30 & 41 & 34 & 18 & 6 \\ 1 & . & . & . & . & . & . & 1 \\ 2 & . & . & . & . & . & . & 1 \\ 3 & . & . & . & . & . & 1 & 1 \\ 4 & . & . & . & . & . & 1 & 1 \\ 5 & . & . & . & . & . & 2 & 1 \\ 6 & . & . & . & . & 1 & 2 & 1 \\ 7 & . & . & . & . & 1 & 3 & . \\ 8 & . & . & . & . & 2 & 3 & . \\ 9 & . & . & . & . & 4 & 2 & . \\ 10 & . & . & . & 1 & 6 & 3 & . \\ 11 & . & . & . & 4 & 11 & 1 & . \\ 12 & . & . & . & 16 & 2 & . & . \\ 13 & . & . & 10 & 1 & 2 & . & . \\ 14 & . & 2 & . & 2 & 2 & . & . \\ 15 & . & . & . & 3 & 1 & . & . \\ 16 & . & . & 1 & 5 & 2 & . & . \\ 17 & . & . & 4 & 9 & . & . & . \\ 18 & . & . & 15 & . & . & . & . \\ 19 & . & 10 & . & . & . & . & . \\ 20 & 2 & . & . & . & . & . & . \end{bmatrix}$$

$$D^4_{4,4} : \quad 0@>>>R(-16)@>>>R^4(-12)@>>>R^8(-6)@>>>R^4(-4)$$

$$D^4_{4,5} : \quad 0@>>>R(-20)@>>>R^4(-15)@>>>R^6(-6)@>>>R^4(-5)$$

$$D^{5,4}_4 : \quad 0@>>>R(-20)@>>>R^4(-15)@>>>R^6(-6)@>>>R^4(-4)$$

$$C_7$$

$$\begin{array}{cc}
 & \begin{array}{cccccccc} 2 & 17 & 65 & 124 & 126 & 72 & 26 & 7 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \\ 19 \\ 20 \\ 21 \\ 22 \\ 23 \\ 24 \\ 25 \end{array} &
\begin{bmatrix}
. & . & . & . & . & . & . & 1 \\
. & . & . & . & . & . & . & 1 \\
. & . & . & . & . & . & 1 & 1 \\
. & . & . & . & . & . & 1 & 1 \\
. & . & . & . & . & . & 2 & 1 \\
. & . & . & . & . & 1 & 2 & 1 \\
. & . & . & . & . & 1 & 3 & 1 \\
. & . & . & . & . & 2 & 3 & . \\
. & . & . & . & . & 3 & 3 & . \\
. & . & . & . & 1 & 4 & 3 & . \\
. & . & . & . & 1 & 5 & 4 & . \\
. & . & . & . & 2 & 8 & 3 & . \\
. & . & . & . & 4 & 34 & 1 & . \\
. & . & . & . & 67 & 3 & & . \\
. & . & . & 43 & 3 & 2 & & . \\
. & . & 8 & 7 & 3 & 2 & & . \\
. & 1 & 7 & 1 & 3 & 2 & & . \\
. & 1 & . & 1 & 4 & 3 & & . \\
. & . & . & 2 & 6 & 2 & & . \\
. & . & . & 3 & 32 & . & & . \\
. & . & . & 66 & . & . & & . \\
. & . & 43 & 1 & . & . & & . \\
. & 8 & 7 & . & . & . & & . \\
1 & 7 & . & . & . & . & & . \\
1 & . & . & . & . & . & & .
\end{bmatrix}
\end{array}$$

Chapter 4

Operations

In this chapter we discuss the algorithms for

membership problems, section 4.1;

elimination and Zariski closure, section 4.2;

homogeneization, section 4.3;

intersection, section 4.4;

transporter, section 4.5;

saturation, section 4.6;

multiple intersections and other multiple problems, section 4.7.

4.1 Membership problems

Definition 4.1 *Let I be an ideal of the ring $R := k[x_1, \ldots, x_n]$ and $f \in R$. The question $f \in R$? is the* membership problem *for I and f. The membership problems for modules etc. are expressed in the same way.*

Membership problems are solvable with Gröbner basis techniques.

Proposition 4.2 *Let $R := k[x_1, \ldots, x_n]$, $M \subset R^r$ an R module and $\mu \in \mathrm{TO}(R^r)$. Then a form $F \in R^r$ belongs to M if and only if* $\mathrm{NF}_{\,\mu}(F, M) = 0$

Since to solve a membership problem it is in general necessary to compute a Gröbner basis, membership problems may be quite expensive.

Remark 4.3 *A special case of the membership problem is the testing of the belonging of 1 to an ideal. If 1 belong to the ideal, the it is in the Gröbner basis, hence to test $1 \in I$ we can execute the Buchberger algorithm on I using any term-ordering, and stop the computations when we found a pair the reduces to 1. Since we don't compute the whole Gröbner basis, the savings may be substantial.*
Since $1 \in I \Longleftrightarrow \mathbf{V}(I) = \emptyset$, this test is quite useful and performed often, for example testing the belonging of a polynomial to the radical of some ideal. This method may be applied any time we know that if a polynomial belongs to an ideal it belongs to its Gröbner basis.

Proposition 4.4 *Let $R := k[x_1, \ldots, x_n]$, I an ideal of R and $f \in R$. Then $f \in \sqrt{I}$ if and only if $1 \in (I, tf - 1) \subset R[t]$.*

4.2 Elimination and Zariski closure

Definition 4.5 *Let $I := f_1, \ldots, f_r \subset k[x_1, \ldots, x_n]$ and $\mathcal{S} := \{x_j\}_{j \in S \subset \{1 \ldots n\}}$. Then the* elimination ideal *$I_{\mathcal{S}}$ is the ideal $I \cap k[\{x_1, \ldots, x_n\} - \mathcal{S}]$*

Remark 4.6 *The ideal $I_{\mathcal{S}}$ is the ideal of the relations among the variables in S.*

Theorem 4.7 *Let $I := f_1, \ldots, f_r \subset k[x_1, \ldots, x_n]$, $\mathcal{S} := \{x_j\}_{j \in S \subset \{1 \ldots n\}}$ and τ be an elimination ordering for $\mathcal{S}$. Let $G'_{\mathcal{S}}$ be a Gröbner basis of I with respect to τ. Then*

$$G_{\mathcal{S}} := G_{\mathcal{S}} \cap k[\{x_1, \ldots, x_n\} - \mathcal{S}]$$

is a Gröbner basis of $I_{\mathcal{S}}$ with respect to τ, the elimination Gröbner basis *for the variables in $\mathcal{S}$.*

Proof. See, for example, [CLS]. □

The computation of elimination Gröbner bases is one of the most basic steps in the effective computation of ideal/module operations, as we see in this chapter.

For ease of notation, it is handy to introduce the definition:

Definition 4.8 *If $\mathcal{S} = \{x_1, \ldots, x_k\}$, then $I_k := I_{\mathcal{S}}$ and $G_k := G_{\mathcal{S}}$.*

Theorem 4.9 *Let I be an ideal of $R := k[x_1, \ldots, x_n]$. Let $V := \mathbf{V}(I \subset k^n)$ and $\pi_k : k^n @ >>> k^{n-k}$ the projection on the last $n-k$ components. Let I_k be the k^{th} elimination ideal of I. Then $\mathbf{V}(I_k)$ is the Zariski closure of $\pi_k(V)$*

Proof. See, for example, [CLS]. □

The Hilbert-Driven algorithm is especially useful in elimination problems, since in several interesting cases for some term-ordering the initial system of generators is already a Gröbner basis. This is the case for

- the computation of the equation of a curve starting from the parameterization of the curve;
- the computation of the equation of a polynomial algebra;
- the computation of the Rees algebra of an ideal.

and several other cases of interest. For some example, see 2.6.2.

Let I be an ideal of the ring $R := k[x_1, \ldots, x_n]$ graded by the system of weights $\mathbf{w}$. We want to compute the elimination ideal I_k. If we know the Hilbert function of R/I, we may use the Hilbert-Driven algorithm. If we know the Hilbert function of R/I_k, we may use a variant of the Hilbert-Driven algorithm that eliminates only the pairs in $k[x_{k-1}, \ldots, x_n]$. If we know both Hilbert functions, we may use them both at the same time. In the discussion on transporters and saturations we have some example of this situation. Here is the code for this double Hilbert-Driven algorithm. The double Bi-Hilbert-Driven algorithm is left to the reader. All considerations are the same as in the discussion of the Hilbert-Driven algorithm, ref. 2.5.

Let $G_k := G \cap k[x_{k+1}, \ldots, x_n]$.

Input $G := \{g_1, \ldots, g_r\} \subseteq k[\mathbf{x}]$, with g_i $\mathbf{w}$ homogeneous
$\tau \in \mathrm{TO}(R)$ such that $\tau = \sigma_{x_1, \ldots, x_k} \times \sigma x_{k+1}, \ldots, x_n$
$\mathbf{w}$, system of weights for R
$\langle (G) \rangle^{\mathbf{w}}$
$\langle (G_k) \rangle^{\mathbf{w}}$
Output the reduced Gröbner basis of G_k with respect to τ.

begin
$L :=$ generates the initial pairs form G;
$\mathbf{d} := \partial_{\mathbf{w}}(\mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w}} - \langle (G)\rangle^{\mathbf{w}}))$;
$\mathbf{d}' := \partial_{\mathbf{w}}(\mathrm{in}(\langle \mathrm{Lt}_\tau(G_k)\rangle^{\mathbf{w}} - \langle (G_k)\rangle^{\mathbf{w}}))$;
Kill the pairs in L whose **w** degree is lesser than **d**;
Kill the pairs in L that belong to L^k and whose **w** degree is lesser than $\mathbf{d}'$;
while $L \neq \emptyset$ **do begin**
choose $(g_i, g_j) \in L$ s.t. $\mathbf{d} = \partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j))$;
$L{:=}L - \{(g_i, g_j)\}$;
$f := \mathrm{NF}\ _\tau(\mathrm{SPoly}\,(g_i, g_j), F)$;
if $f \neq 0$ **then** $F'{:=}F' \cup \{h\}$;
if $\mathbf{d} \neq min_{\mathbf{w}}\{\partial_{\mathbf{w}}(\mathrm{SPoly}\,(g_i, g_j)) \mid (g_i, g_j) \in L\}$ **then**
begin
$G'{:=}Interreduce(G')$;
L,G:=$\mathtt{Upgrade}^{\mathbf{w},\mathbf{d}}(L, G', G)$;
$L'{:=}\emptyset$;
$\mathbf{d} := \mathrm{in}(\langle \mathrm{Lt}_\tau(G)\rangle^{\mathbf{w}} - \langle (G)\rangle^{\mathbf{w}})$;
$\mathbf{d}' := \partial_{\mathbf{w}}(\mathrm{in}(\langle \mathrm{Lt}_\tau(G_k)\rangle^{\mathbf{w}} - \langle (G_k)\rangle^{\mathbf{w}}))$;
endif;
Kill the pairs in L whose **w** degree is **w** lesser than **d**;
Kill the pairs in L that belong to L^k and whose **w** degree is lesser than $\mathbf{d}'$;
endwhile;
return G;
end.

This double Hilbert-Driven algorithm has several variants, depending on the knowledge on the input polynomials. We may know the Hilbert Function of the input ideal with respect to some system of weights **w**, and the Hilbert Function of the elimination ideal with respect to some other system of weights **u**, and so on and so forth. The variants are all quite easy to develop from the algorithms we have shown.

4.3 Homogeneization

Definition 4.10 *Let $R := k[x_1, \dots, x_n]$, $h_1, \dots, h_k$ be new variables and I be an ideal of R. Then the couple $(I^h, \mathbf{w})$, where $I^h \subset R[\mathbf{h}]$ and $\mathbf{w}$ is a system of weights on $R[\mathbf{h}]$ is an homogeneization of I if*

a) I^h is $\mathbf{w}$ homogeneous;

b) $\{f(x_1, \dots, x_n, 1, \dots, 1) | f \in I^h\} \subset R$ generates I.

Theorem 4.11 *Let $R := k[x_1, \dots, x_n]$, I be an ideal of R and $(I^h, \mathbf{w})$ be an homogeneization of I. Let $\tau \in \mathrm{TO}(R)$ be compatible with $\mathbf{w}_{|R}$ and G be the Gröbner basis of I with respect to τ. Then the homogeneization of G with respect to $\mathbf{w}$ is a Gröbner basis for I^h with respect to a term-ordering τ' compatible with $\mathbf{w}$.*

Proof. The proof for type one systems of weights is standard, see e. g. [CLS]. The generalization to higher type systems of weights is immediate. □

Remark 4.12 *Theorem 4.11 gives an immediate algorithm for the computation of $I^{h,\mathbf{w}}$: first compute the Gröbner basis for a term-ordering compatible with $\mathbf{w}$, the homogenize it with respect to $\mathbf{w}$.*

Proposition 4.13 *Let $R := k[x_1, \dots, x_n]$ and I be an ideal of R. If $(I^h, \mathbf{w})$ is an homogeneization of I and $\tau \in \mathrm{TO}(R[\mathbf{h}])$ is compatible with $\mathbf{w}$, then the de-homogeneization of $\mathrm{GB}(I^h, \tau)$ gives a Gröbner basis of I with respect to τ_R.*

Proof. Immediate. □

Remark 4.14 *Theorem 4.11 allows us to compute the Gröbner basis of an homogeneization of a non-homogeneous ideal I. In several instances we have a non-homogeneous ideal whose generators, if homogenized would give a simpler Gröbner basis, or allow to use easily the Hilbert-Driven algorithm. In these cases, homogeneizing the generators may be worthwhile. We will see some example in the use of the Hilbert function in intersections, transporters and saturations.*

Proposition 4.15 *Let $R := k[x_1, \ldots, x_n]$, $I := (f_1, \ldots, f_l)$ be an ideal of R and $\tau \in \mathrm{TO}(R)$. Let $\mathbf{h}$ be new variables, $\mathbf{w}$ be a system of weights for $R[\mathbf{h}]$ and $\{f_1^{\boldsymbol{h},\mathbf{w}}, \ldots, f_l^{\boldsymbol{h},\mathbf{w}}\}$ be homogeneizations of $f_1, \ldots, f_l$. Then a Gröbner basis of $(f_1^{\boldsymbol{h},\mathbf{w}}, \ldots, f_l^{\boldsymbol{h},\mathbf{w}})$ with respect to an elimination ordering for $\mathbf{x}$ is a Gröbner basis for I if we substitute the variables $\mathbf{h}$ with $1, \ldots, 1$. Moreover, in the Buchberger algorithm, if it appears a polynomial whose leading term doesn't have powers of the $\mathbf{x}$, we have that $I = (1)$.*

Proof. Since we are adding variables that we be then valued as 1, the Gröbner basis of $(f_1^{\boldsymbol{h},\mathbf{w}}, \ldots, f_l^{\boldsymbol{h},\mathbf{w}})$ after the valuation generates I. If a polynomial whose leading term is a pure power of $\mathbf{h}$ appears in the Gröbner basis, then after the valuation $I = (1)$. Since we are using an elimination term-ordering for the $\mathbf{x}$ variables, the Gröbner basis is still a Gröbner basis after the valuation. □

Remark 4.16 *Proposition 4.15 allows us to compute Gröbner basis of non-homogeneous ideals using the homogeneous algorithm (and maybe the Hilbert-Driven or Bi-Hilbert-Driven algorithms). The advantages are almost always offset by the increased complexity of the computation due to the introduction of additional variables (hence, components at infinity of the variety).*

4.4 Intersection

Intersection is another of the most basic steps. Here we describe the two classical algorithms to compute submodule intersections, together with some less known variants.

The tag variable algorithm

The algorithm relies on the

Theorem 4.17 *Let $R := k[x_1, \ldots, x_n]$ and N, H be submodules of a free R-module M. Let t be a new indeterminate, and consider the $R[t]$ modules M_t, N_t, H_t, $M_t = M \otimes_R R[t]$, etc.; a presentation of N, H lifts to a presentation of N_t, H_t. Then*

$$(tN_t + (1+t)H_t) \cap M = N \cap H$$

Proof. An (incomplete) proof is the following: consider $\gamma = t\alpha + (1+t)\beta$, and assume $\gamma \in M$, $\alpha \in N$, $\beta \in H$. Since $\gamma = \beta + t(\alpha+\beta)$, and t does not appear in γ, this means that $\alpha+\beta = 0$ (hence $\alpha = -\beta \in H$) and $\gamma = \beta = -\alpha \in N \cap H$. The converse inclusion is straightforward. □

The proof is incomplete, since we cannot a priori assume $\alpha \in N$, but only $\alpha \in N_t$. The proof can anyway be adjusted (induction on the degree of t in α; the proof above is the level 0 of induction). The present "proof" proves something else that will be revealed in a next section.

This algorithm is hence a special case of elimination, and hence can be performed by the Buchberger algorithm with respect to an elimination term-ordering (see [Sp], [GTZ], etc.).

The syzygy algorithm

The syzygy algorithm is based on the following theorem :

Theorem 4.18 *Let $R := k[x_1, \dots, x_n]$, $N, H \subset R^r$ be R modules and the sequences*

$$R^r \to N \to 0 \qquad R^s \to H \to 0$$

be exact; hence the induced map

$$R^{r+s} \to N + H \to 0$$

is exact. Let

$$R^l \to^{\phi} R^{r+s} \to^{\psi} N + H \to 0$$

be an exact extension. Then let $\pi : R^{r+s} \to R^r$, $\iota : R^r \to R^{r+s}$ be the canonical projection and injection. Then

$$\psi\iota\pi\phi(R^l) = N \cap H$$

Proof. The proof is almost immediate, see [E]. To find explicitly these relations, see [Sh], [MM], [BW], or a subsequent section. □

Hence to find the intersection $N \cap H$ one can find the relations between the generators of N and H (e. g. the map ϕ).

Remark 4.19 *Let $R := k[x_1, \dots, x_n]$ and $H = (h_i)$ and $N = (n_j)$ be submodules of the R module M. The syzygy algorithm to compute the intersection between H and L computes the relations between the (h_i, n_j) and then for each generator γ of the module of relations takes the scalar product $\langle \gamma, (h_i, 0) \rangle$.*

A third algorithm

In [E], Exercise 15.42, a third algorithm is reported (see also [BGS] if you can): compute the first coordinate of the kernel of the map defined by the matrix

$$\begin{pmatrix} 1 & f_1 & \dots & f_s & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & g_1 & \dots & g_t \end{pmatrix}$$

where $(f_i) = I$ and $(g_j) = J$. The proof of the correctness of the algorithm is immediate. The algorithm allows us obvious extensions to the case of submodules of a free module M (the map being in that case a map to $M \oplus M$, and the initial 1 of each line being interpreted as a square matrix).

These algorithms have variants for multiple intersections. Assume that we want to compute $I_1 \cap \dots \cap I_n$. The tag variable algorithm adds variables $t_1, \dots, t_n$, with the relation $t_1 + \dots + t_n + 1$, and eliminates the t_i from $t_1 I_1 + \dots + t_n I_n$. The third algorithm adds further lines with the generators of the other ideals.

4.4.1 Intersections in practice

We want to show that the syzygy algorithm, the tag variable algorithm and the third algorithm are essentially the same.

In the syzygy algorithm, assume that the syzygies are only used to compute the intersection, and then discarded. We show how to minimize the computations and the storage.

To avoid the final computation, one can simply:

a) avoid considering the components of the syzygies that correspond to the n_j, since they are anyway discarded at the end.

b) perform at the very start the scalar product that is needed at the end. This consists in applying the distributive property of the scalar product.

Item a) is obviously advantageous, since we avoid to perform computations whose result is at the end multiplied by zero; the convenience of b) might be questioned —we consider a polynomial instead of a vector of polynomials, but the single polynomial might be more complicated that the vector— but in practice this is not the case.

In matrix form, the original algorithm would compute the Gröbner basis of the matrix

$$\begin{pmatrix} f_1 & \dots & f_s & g_1 & \dots & g_r \\ 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}$$

while after improvement a) we have to compute the Gröbner basis of the matrix

$$\begin{pmatrix} f_1 & \dots & f_s & g_1 & \dots & g_r \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}$$

and after improvement b) we compute just the Gröbner basis of the two-rows matrix

$$\begin{pmatrix} f_1 & \dots & f_s & g_1 & \dots & g_r \\ 0 & \dots & 0 & g_1 & \dots & g_r \end{pmatrix}$$

I. e.the Gröbner basis of

$$\{(t+1)f_1, \dots, (t+1)f_s, t\,g_1, \dots, t\,g_r\} \;=\; (t+1)H + tN \subseteq t\,k[\mathbf{x}] \oplus k[\mathbf{x}]$$

The usual tag variable intersection algorithm do the same computation, but the tag variable t is a ring variable (*ring setting*). But indeed, looking closely at the tag variable intersection algorithm, we can improve it: if the tag variable t is considered to be a "module variable", (*module setting*) in the sense of a preceding section, the algorithm still computes the intersection (the incomplete proof given in a previous section now is correct!) and the algorithm is simpler: Since we are using the module Buchberger algorithm, the pairs (f', f'') such that $\mathrm{Lt}(f')$ contains t and $\mathrm{Lt}(f'')$ are not considered.

We have so reconciled the two algorithms. Both can be improved to become coincident. Indeed,

> in the tag variable algorithm, if we are not interested to have an answer that is Gröbner, we may discard all pairs of elements that do not contain the tag variable t.
>
> In the syzygy algorithm, if we want to have an answer that is Gröbner, we have to perform a Buchberger algorithm on the result.

Remark 4.20 *There is still a difference between the two, in the timings of the different phases: if we first completely compute syzygies, and then a Gröbner basis of the intersection, this can be seen as interrupting the simplification phase of the Buchberger algorithm and resuming it later. Whether this is advantageous or not is not yet clear, and further experiments might be needed to settle the issue.*

Just a few comments on the "third algorithm". We have to compute the kernel of a map, e. g. the syzygies of the columns of a matrix, then take the first component of those syzygies. We can hence avoid to compute these components, as we have proposed for the first improvement of the syzygy algorithm. Hence we have to compute the Gröbner basis of the module generated by the columns of the matrix

$$\begin{pmatrix} 1 & f_1 & \dots & f_s & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & g_1 & \dots & g_t \\ 1 & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix}$$

To compute the module Gröbner basis, we multiply to the left by $(t, s, 1)$, t, s being module variables; then eliminate t and s. The first column is $t + s + 1$, the others are generators of tI and sJ respectively; this means that we are performing the multiple intersection tag-variable algorithm.

4.4.2 An interpretation

The tag variable algorithm, in the module setting, has an interpretation that provides another proof of the correctness of the algorithm.

Remark 4.21 *let A be a ring, then consider $A \oplus A$ as A-module and its submodules $M_1 = \{(a, a) \mid a \in I\}$, $M_2 = \{(b, 0) \mid b \in J\}$ and $M = M_1 + M_2$. Then*

$$M \cap (0 \oplus A) = 0 \oplus (I \cup J)$$

and this is exactly what the tag variable algorithm (module setting) computes.

4.4.3 Useless pair avoidance

Assume that we have to compute the intersection of two ideals $I, J \subset R := k[x_1, \dots, x_n]$ of which we have a Gröbner basis, say $F = \{f_i\}$ and $G = \{g_j\}$.

When computing a Gröbner basis with input $H = tF \cup (t+1)G$ we have to consider the critical pairs of elements of H. But we can remark that the pairs (tf_i, tf_h) will surely be simplified to 0 by elements of tF, exactly in the same way as (f_i, f_h) is simplified to 0 by elements of F. In the same way the pairs $((t+1)g_j, (t+1)g_h)$ will be simplified to 0 by elements of $(t+1)G$.

Thus an optimized implementation of intersection can take advantage of this remark. The same remark can be useful for the syzygy algorithm, if we know the syzygies of F and G; but apparently this is more difficult and of less use.

Remark 4.22 *Using either algorithm, if we don't need a Gröbner basis of the intersection module, we may discard a pair if the leading terms of both its element that don't contain the variable t.*

We show now that, although it seems completely equivalent to the tag variable algorithm, the "third algorithm" does not allow the aforementioned optimization.

Consider the module generated freely by $t, s, 1$ (with an ordering eliminating t and s, $t > s$) and the submodule M generated by $tf_i, sg_j, t+s+1$; to compute a Gröbner basis of M we can avoid all the critical pairs between elements of tf_i, and all the critical pairs between the sg_j; these elements moreover do not build critical pairs since they refer to different components; we have to start computing the critical pairs between the tf_i and $t + s + 1$. Each of these pairs generates an element $-(s+1)f_i$, hence we are again to the standard form of the tag-variable module algorithm. But in this preliminary seemingly innocuous step we have lost the information that the pairs $((s+1)f_i, (s+1)f_j)$ go to 0, e. g. the information that we start from a Gröbner basis of I.

If we start from a set of generators that are not a Gröbner basis the drawback is similar, since we first compute a Gröbner basis of tI just to discard the knowledge that it is a Gröbner basis recomputing it divided by t and multiplied by $(s+1)$ instead.

Proposition 4.23 *Let $R := k[x_1, \dots, x_n]$, and let $M, N \subset R^r$ be R modules. Then, using the matrix notation for the modules,*

$$\text{the Gröbner basis of } \begin{bmatrix} M & N \\ \mathbf{0} & N \end{bmatrix} \text{ with respect to } \mu \text{ is } \begin{bmatrix} \mathrm{GB}(M+N, \mu_{|R^r}) & \mathbf{0} \\ \mathbf{F} & \mathrm{GB}(M \cap N, \mu_{|R^r}) \end{bmatrix}$$

if μ is a block module term-ordering on $R^r \oplus R^r$ for which the first component is the biggest.

Proof. The first component of the Gröbner basis is $\mathrm{GB}(M+N, \mu_{|R^r})$ since μ is a block ordering for the first component. That the second component of the Gröbner basis is $\mathrm{GB}(M \cap N, \mu_{|R^r})$ follows from the proof of the tag variable intersection algorithm. □

Remark 4.24 *If I is an ideal of $R := k[x_1, \dots, x_n]$, t is a new variable and τ is an elimination ordering for t on $R[t]$, then $GB((I, (1+t)I), \tau) = GB(I, \tau_{|R})$. If t is a module variable, then*

$$GB((I, (1+t)J), \tau) = \begin{bmatrix} GB(I+J, \tau_{|R}) & 0 \\ F & GB(I \cap J, \tau_{|R}) \end{bmatrix}$$

Remark 4.25 *Using the tag variable module algorithm, the computation of $M \cap N$ has as side effect the computation of a Gröbner basis of $M + N$.*

4.4.4 Hilbert-Driven intersection

If the ideals I and J are homogeneous with respect to a system of weights $\mathbf{w}$, then the tag variable intersection algorithms may be easily adapted to keep the homogeneity. It is only necessary to give to the tag variable a suitable weight.

Proposition 4.26 *Let $R := k[x_1, \dots, x_n]$ and $I, J \subset R$ be $\mathbf{w}$-graded ideals. Let t_1, t_2 be new variables. Then*

1. *the ideal $(t_1 I, (t_1 + t_2)J) \subset R[t_1, t_2]$ is graded with respect to the system of weights*

$$\mathbf{u} := \left[\begin{array}{c|cc} & 1 & 1 \\ [\mathbf{w}] & \vdots & \vdots \\ & 1 & 1 \end{array} \right]$$

2. *Let τ be a term-ordering on $R[t_1, t_2]$ such that*
   a. *it is an elimination ordering for t_1.*
   b. *the variable t_2 is block-lesser than the other variables.*

   Then $(GB(t_1 I, (t_1 + t_2)J, \tau))_{|t_2=1} \cap R = I \cap J$.

Proof. Immediate. □

Proposition 4.27 *Let $R := k[x_1, \dots, x_n]$ and $M, N \subset R^r$ be $\mathbf{w}$ graded R modules. Let t_1, t_2 be a new module variables. Then*

1. *The module $(t_1 M, (t_1 + t_2)N) \subset t_1 R^r \oplus t_2 R^r$ is graded with respect to the system of weights*

$$\mathbf{u} := \left[\begin{array}{c|cc} & 1 & 1 \\ [\mathbf{w}] & \vdots & \vdots \\ & 1 & 1 \end{array} \right]$$

2. the module $(t_1 M, (t_1 + t_2)N)$ is $\mathbf{w}$ graded isomorphic to the module $(tM, (t+1)N)) \subset t\,R^r \oplus R^r$.

Proof. Immediate from proposition 4.27. $\square$

Corollary 4.28 *If M, N are $\mathbf{w}$ graded, then*

$$\mathcal{H}^{\mathbf{u}}_{(t_1 M + (t_1+t_2)N)} = \mathcal{H}^{\mathbf{w}}_{M+N} + \mathcal{H}^{\mathbf{w}}_{M\cap N} = \mathcal{H}^{\mathbf{w}}_{M\oplus N}$$

Proof. Follows immediately from the exactness of the sequence in proposition 1.111. $\square$

Remark 4.29 *If we have two ideals I and J, we may consider them as modules and use the tag-variable algorithm with module tag variables. We see in the examples of section 4.4.5 that this trick may be useful. If we happen to know the Hilbert function of $M+N$ or $M \cap N$ we can skip even more pairs, see subsection 4.2.*

The use of the Hilbert-Driven or Bi-Hilbert-Driven algorithm is then immediate for the intersection of graded ideals/modules. If M or N are not homogeneous, then they can be homogenized, see section 4.3. This is usually a good idea since the costs for the computation of the Gröbner bases of M and N are usually quite small in comparison to the cost for the computation of $M \cap N$.

4.4.5 Intersection statistics

There are no standard benchmarks for intersection. To have examples coming from some practical problem, we intersect the ideals produced by the pure dimensional ideal decomposition algorithm of chapter 5. We choose four of the examples introduced in section 5.5. In every case, the ideal is homogenized, and then the decomposition algorithm is executed. The lists have

Gerdt 5 ideals.

T2 2 ideals.

Butcher 19 ideals.

Oaku 6 ideals.

The ideals are then intersected.

Headings :
HMod is the Hilbert-Driven module algorithm, **Mod** is the module algorithm, **HRing** is the Hilbert-Driven ring algorithm, **Ring** is the ring algorithm and **3** is the "third" algorithm. The timings are for the computation of the intersection of all the ideals in the lists using the naive multiple intersection algorithm.

Name	HMod	Mod	HRing	Ring	3
Gerdt	4.70 s	12.70 s	5.30 s	10.10 s	13.00 s
T2	0.3 s	0.70 s	0.2 s	0.80 s	0.3 s
Butcher	10.80 s	22.80 s	13.10 s	15.60 s	24.10 s
Oaku	0.50 s	0.50 s	0.70 s	0.50 s	0.40 s

Here there are some other tests : we get an ideal each from the decomposition of the ideals Gerdt, T2, Butcher and Oaku. Then we intersect them. We see the data intersection by intersection.

Name	HMod	Mod	HRing	Ring	3
$G \cap T$	0.50 s	2.30 s	0.50 s	1.60 s	2.20 s
$G \cap T \cap B$	6.20 s	8.30 s	6.40s	7.60 s	8.30 s
$G \cap T \cap B \cap O$	70.80 s	164.20 s	190.00s	245.60 s	1175.80 s

We note that the tag variable module Hilbert-driven algorithm is clearly the best.

Some remarks on the computations :

1. in every case, it is slightly better to start the intersection algorithms with the Gröbner bases as the system of generators of the ideals.

2. The algorithms compute different Gröbner bases, but the Gröbner bases differ very little in the number of polynomials. Usually the ring algorithm basis has one or two less polynomials than the module algorithm.

3. The computation of the Gröbner bases of the ideals has in every instance negligible costs with respect to the costs of the intersection algorithms.

4. The timings are from a prototypical high-level implementation of the algorithms, hence its only interest is for an evaluation of the algorithms.

4.5 Transporter

The algorithms for the computation of the transporter have apparently more variety than the algorithms for the intersection. These algorithms are not much different for the ideal and the module case, so we describe just the ideal case, e. g. we compute $I : J$, $I = (f_1, \dots, f_n)$, $J = (g_1, \dots, g_m)$, where I, J are ideals of $R := k[x_1, \dots, x_n]$.

We can classify these algorithms in three categories:

1. Algorithms when J is homogeneous and generated by the last variable in DegRevLex ordering:
   (a) Special remark by Bayer.
2. Algorithms when J is principal
   (a) Divided intersection algorithm.
   (b) Tag variable algorithm.
   (c) Syzygy algorithm.
   (d) Reduction to the special remark by Bayer (homogeneous ideals only).
3. Algorithms when J has many generators
   (a) Through intersection.
   (b) Through random elements.
   (c) Through a "special generic element".

We want to show that, notwithstanding the number of these algorithms, they are just the same algorithm with small variations.

4.5.1 Bayer special remark

Assuming that I is homogeneous and J is generated by the last variable x_r in DegRevLex ordering, the ideal $I : J$ is obtained dividing by the last variable all the elements of a Gröbner basis of I whose head is divisible by x_r (by the definition of DegRevLex, if f is homogeneous and its head is divisible by x_r then f itself is divisible by x_r).

4.5.2 Principal quotient

Let $I := (f_1, \dots, f_k)$ and $J := (g)$.

Divided intersection algorithm $I : J$ is obtained dividing by g all the elements of $I \cap (g)$.

Tag variable module algorithm $I : J$ is obtained eliminating the additional (module) variable t from $tI + (tg + 1)$. This algorithm allows us a variant: eliminate t from $tI + I + (tg + 1)$.

Syzygy algorithm Consider the syzygies of $(g, f_1, \dots, f_n)$ and consider the first component.

Reduction to the special remark by Bayer If the ideals I and $J := (g)$ are homogeneous, we add a new variable y of degree $\deg(g)$, and then we compute $(I, g - y) : (y)$ with the Bayer remark algorithm and then eliminate y from the result.

Proof. The proofs of these facts are all standard, see e. g. [CLS], except for the Tag variable module algorithm. We have to prove that $I : (g) = (tI, tg + 1) \cap R$. But $(tI, tg + 1) \cap R = \{t \sum h_i f_i + h(tg + 1) \mid h_i, h \in R\} \cap R = \{h \in R \mid hg = -\sum h_i f_i\} = \{h \in R \mid hg \in I\} = I : (g)$ □

The divided intersection algorithm is equal to the tag variable algorithm or to the syzygy algorithm, depending on the algorithm of your choice for the intersection, with the only difference that one component of all the polynomials is multiplied by g before the algorithm and divided by g at the end. This makes all the computations slower, but the algorithm does not change. We have already seen that the intersection algorithms are the same, hence the three algorithms are substantially the same.

The syzygy algorithm can of course be improved avoiding to compute the components of the syzygies that are discarded at the end; we have hence to compute the Gröbner basis of

$$\begin{pmatrix} f_1 & \dots & f_n & g \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

and is hence identical to the tag variable algorithm (if t is taken, as it should, as module variable).

The variant of the tag variables algorithm is convenient if we start from a Gröbner basis: the addition of the generators of I does not require the addition of further critical pairs, and can avoid many computations. For example, when $I : (g) = I$, adding initially a Gröbner basis f_i of I avoids at the end to prove that this same set f_i is a Gröbner basis.

Proposition 4.30 *Let $R := k[x_1, \dots, x_n]$, I be an ideal of R and $g \in R$. Then, using the matrix notation for the modules,*

$$\text{the Gröbner basis of } \begin{bmatrix} I & g \\ \mathbf{0} & 1 \end{bmatrix} \text{ with respect to } \mu \text{ is } \begin{bmatrix} \mathrm{GB}((I,g), \mu_{|R}) & \mathbf{0} \\ \mathbf{F} & \mathrm{GB}(I : (g), \mu_{|R}) \end{bmatrix}$$

if μ is a block module term-ordering on $R \oplus R$ for which the first component is the biggest.

Proof. The first component of the Gröbner basis is $\mathrm{GB}(I, g)$ since the μ is an elimination ordering for the first component. That the second component of the Gröbner basis is $\mathrm{GB}(I : (g))$ follows from the proof of the tag variable module algorithm. □

Remark 4.31 *The computation of $I : (g)$ with the tag variable module algorithm has as side effect the computation of a Gröbner basis of (I, g); this may be quite useful, see chapter 5. If we happen to know the Hilbert function of (I, g) or $I : (g)$, we have the same possibilities as in the Hilbert-Driven intersection algorithm.*

4.5.3 Principal quotient - graded case

If the ideals I and J are homogeneous with respect to a system of weights $\mathbf{w}$, then the transporter algorithms may be easily adapted to keep the homogeneity. It is only necessary to give to the tag variables suitable weights.

For the divided intersection algorithm, follow section 4.4.4. For the syzygy algorithm, the Hilbert-Driven algorithm may be used forthwith. The tag variable algorithm has to be treated with a bit more care.

Proposition 4.32 *Let $R := k[x_1, \ldots, x_n]$ be graded by the system of weights $\mathbf{w}$, I be a $\mathbf{w}$ homogeneous ideal of R and $g \in R$ $\mathbf{w}$ homogeneous. Let t_1, t_2 be new module variables. Then*

1. *The module $(t_1 I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1}) \subset t_1\, R \oplus t_2\, R$ is graded with respect to the system of weights*

$$\mathbf{u} := \left[\begin{array}{c|cc} & 1 & 1 \\ [\boldsymbol{w}] & \vdots & \vdots \\ & 1 & 1 \end{array}\right]$$

2. *If τ is a term-ordering on $R[t_1, t_2]$ fulfilling the conditions of proposition 4.26, 2. Then*

$$(GB(t_1 I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1}, \tau))_{|t_2=1} \cap R = I \cap J$$

Proof. Immediate from 4.30. □

Corollary 4.33 *With the notation of proposition 4.30 we have*

$$\mathcal{H}^{\mathbf{u}}_{(t_1 I, (t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1}))} = \mathcal{H}^{\mathbf{w}}_{(I,g)} + \mathcal{H}^{\mathbf{w}}_{I:(g)} = \mathcal{H}^{\mathbf{w}}_{I}$$

Proof. Follows immediately from proposition 4.30 and formula 1.110. □

Remark 4.34 *In proposition 4.32 we may use more sophisticated system of weights to have a higher type. For example, the system of weights*

$$\mathbf{u} := \left[\begin{array}{ccc|cc} & [\boldsymbol{w}] & & 1 & 1 \\ & & & \vdots & \vdots \\ & & & 1 & 1 \\ 0 & \cdots & 0 & \partial_{\mathbf{w}}(g)+1 & 1 \end{array}\right]$$

is such that the module $(t_1 I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1})$ is $\mathbf{u}$ homogeneous.

Proposition 4.32 allows us to use the Hilbert-driven techniques in the transporter algorithm, graded case. As in the intersection algorithm, homogeneization of the input is quite useful in non homogeneous cases.

4.5.4 Bayer's remark algorithms

The "special remark by Bayer" algorithm just describes what happens when performing the tag variable algorithm in that special situation; if we use the Hilbert-driven Buchberger algorithm; we assume that f_i is already a Gröbner basis. We multiply all the f_i by t, add the equation $tx_r - 1$, then we compute the Gröbner basis; this computation, using the fact that the input is Gröbner, consists in:

1. substitute tx_r with 1 in all the polynomials having tx_r in the head (S-polynomials between tf_i and $tx_r - 1$ when $x_r \mid Lpp(f_i)$).

2. multiply by x_r all the polynomials not having x_r in the head, then substitute tx_r with 1 (S-polynomials between tf_i and $tx_r - 1$ when $x_r \nmid Lpp(f_i)$); this means, put in the result all the polynomials f_i not divisible by x_r unchanged.

3. perform a Buchberger algorithm on the polynomials not having t in the head, just to discover that they form a Gröbner basis.

Just remark that this last step is avoided using the Hilbert function: we already know that no further useful pair exists. The first step is exactly what is done in the "special remark" algorithm, and the second just retrieves f_i first multiplying it by t, then multiplying it by x_r, then dividing it by tx_r. Of course, this algorithm is very convenient if we are in this case.

The "reduction to the special remark" just adds a variable to have a simple computation, then eliminates it; but of course requires two Gröbner basis computations for these two steps. This is just an indirect way to perform the tag variable algorithm, through an intermediate computation with a different ordering. Whether this is convenient or not is matter of thorough experimentation.

Remark 4.35 *If g is not homogeneous, and I is homogeneous, we can either homogenize g or split g into its homogeneous components g_i, and divide by the ideal defined by the g_i.*

4.5.5 Non principal quotient

Reduction to principal through intersection Use the fact that $I : (g_1, \dots, g_m) = (I : g_1) \cap \dots \cap (I : g_m)$. A shortcut is possible: if we realize that $((I : g_1) \cap \dots \cap (I : g_{i-1}))g_i \subseteq I$ we can skip $(I : g_i)$ in the intersection. Note that the check $K \subset (I : g_i)$ may be performed testing if $g_k K \subset I$, see subsection 4.7.

Reduction to principal through random elements Compute $I_b = I : (b)$ where b is a (random) element of J; if $I_b J \subseteq I$ then $I_b = I : J$. The point is that if b is "generic" (whatever this means) then I_b is indeed $I : J$, and the result can be checked: if $I_b J \subseteq I$ then $I_b = I : J$, since in any case $I_b \supseteq I : J$.

Reduction to principal through a "special generic element" Add a new variable y, and consider $b = g_1 + y g_2 + \dots + y^{m-1} g_m$; then $I : (b)$ is generated by elements that do not contain y, and these generate $I : J$. The spirit of the algorithm is that b is sufficiently "generic" to make the previous algorithm work.

Proof. The proof of the first two facts are immediate. For the third, We have to prove that $I : (g_1, \dots, g_k) = I : (g_1 y + \dots + g_k y^k)$. But $I : (g_1 y + \dots + g_k y^k) = \{h \in R \mid h(g_1 y + \dots + g_k y^k) \in R\} = \{h \in R \mid h g_i \in I\} = \{h \in R \mid hJ \subset I\} = I : J$. The proof holds if y is a module variable. □

Now we are left with the case in which J has more than one element. There are three algorithms to consider; one of them (reduction to principal through a random element) is not easy to analyze, just because of the randomness, but can be considered a variant of the "reduction to principal through intersection": b is added as the first generator, and after computing $I : (b)$ we check that the other intersections are not needed (and if not we restart with another random element).

The reduction to principal through a random element is not easy algorithm to analyze, just because of the randomness, but it can be considered a variant of the "reduction to principal through intersection" algorithm: b is added as the first generator, and after computing $I : (b)$ we check that the other intersections are not needed (and if not we restart with another random element). The "reduction to principal through a 'special generic' element" algorithm is on one side a form of "reduction to principal through a random element" algorithm in which the element $b := g_1 + y g_2 + \dots + y^{m-1} g_m$ is proved to be sufficiently random. On the other side we will see that this algorithm has a lot of relations with the "reduction to principal through intersection" algorithm.

To understand it, first modify the algorithm considering the algorithm not in the ring $k[X, y]$ but in the free $k[\mathbf{x}]$-module $M = k[\mathbf{x}, y]/(y^m)$. Instead of I consider $I \otimes M$ (generated by the $f_i \otimes y^j$), and divide it by the element $b = g_1 + yg_2 + \cdots + y^{m-1}g_m$. This means considering the Gröbner basis of the ideal generated by the columns of the matrix

$$\begin{pmatrix} f_1 & \dots & f_n & 0 & \dots & 0 & \dots & 0 & \dots & 0 & g_m \\ 0 & \dots & 0 & f_1 & \dots & f_n & \dots & 0 & \dots & 0 & g_{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & f_1 & \dots & f_n & g_1 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & 1 \end{pmatrix}$$

with respect to an ordering eliminating all the components except the last one.

Computing instead with the "reduction to principal through intersection" we have first to compute several $I : (g_i)$ e.g. through matrices

$$\begin{pmatrix} f_1 & \dots & f_n & g_m \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

then compute the intersection e.g. through the multivariate form of the "third algorithm". Combining all the computations in one, we have to consider the matrix

$$\begin{pmatrix} f_1 & \dots & f_n & 0 & \dots & 0 & \dots & 0 & \dots & 0 & g_m & 0 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & f_1 & \dots & f_n & \dots & 0 & \dots & 0 & 0 & g_{m-1} & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & f_1 & \dots & f_n & 0 & 0 & 0 & \dots & g_1 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

and it is easy to pass from the one to the other.

From the description it is not easy to see which algorithm should be preferred; there are different issues:

- Is it more convenient to use the module or the ring algorithm for the "special generic element" algorithm? While a naive implementation would require a recomputation of the Gröbner basis of I for each component, this of course could be done once and for all. It is however clear that duplications in the intermediate results remain, unless more sophisticated tricks are used. On the other side, for the module algorithm the computation is bounded in degree $< r$ in y (but this can also be obtained adding the generator y^r in the ring case).

- Is it more convenient to collect the results of the different $I : (g_j)$ then intersect them, or to compute the Gröbner basis of the big matrix?

- What is the difference in the practical computation between the smaller and the larger matrix?

- What is the influence of the possibility to use the Hilbert-driven algorithm?

Our guess is that the possibility of using the Hilbert-driven algorithms overwhelms all the possible advantages of the other variants. But this has to be proved in practice.

4.5.6 Transporter statistics

As in the intersection situation, there are no standard benchmarks for transporter. Here there are some examples. Gerdt and Gerdt' come from the example Gerdt of chapter 5. C_6 and C_6' are slight modification of the cyclic 6 ideal. We have tested the "reduction to the Bayer remark" Hilbert-driven algorithm and the tag variable Hilbert-driven module algorithm for the principal quotient case.
Headings: **Bayer** gives the timings for the Hilbert-driven reduction to Bayer remark algorithms, while **Htr** gives the timings for the tag-variable Hilbert-driven module algorithm.

Name	Bayer	HTr
Gerdt	11.7 s	14.9 s
Gerdt'	16.2 s	13.1 s

Name	Bayer	HTr
C_6	48.2 s	12.5 s
C_6'	23.1 s	12.5 s

In the Gerdt example, the ideal was discovered to be (1) after the DegRevLex computation in the Bayer remark algorithm. The Tag variable Hilbert-Driven module algorithm seems to be the best (and more stable). Since the number of algorithms and their possible combinations are high, we are developing an automatic tester. See section 4.36.

4.6 Saturation

There are two algorithms for the computation of a saturation if the saturating ideal is principal.

Proposition 4.36 *Let $R := k[x_1, \dots, x_n]$, I be an ideal of R and $g \in R$. Then*

a) there exists $n \in \mathbb{N}$ such that $I : (g)^n = I : (g)^\infty$.

b) If t is a new ring variable, $(I, tg + 1) \cap R = I : (g)^\infty$.

Proof. See e. g. [CLS]. $\square$

Using the first part of the proposition, a saturation is computed by several transporter computations. Any transporter algorithm may be used. Using the second part of the proposition, a saturation is computed directly via an elimination Gröbner basis. Which algorithm is the best depends from the input, but usually *a)* is better since it leads to split computations.

If I, g are homogeneous with respect to a system of weights $\mathbf{w}$, it is possible to use the graded variants of the transporter algorithms, or the following

Proposition 4.37 *Let $R := k[x_1, \dots, x_n]$ be $\mathbf{w}$-graded, I be a homogeneous ideal of R and g be an homogeneous element of R. Let t_1, t_2 be new ring variables. Then*

a. the ideal $(I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1})$ is homogeneous with respect to the system of weights

$$\mathbf{u} := \left[\begin{array}{c|cc} & 1 & 1 \\ [\boldsymbol{w}] & \vdots & \vdots \\ & 1 & 1 \end{array} \right]$$

b. If τ is a term-ordering fulfilling the conditions of proposition 4.26, 2. then we have

$$(GB(I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1}, \tau))_{|t_2=1} \cap R = I : (g)^\infty$$

c. If $d = \partial_{\mathbf{w}}(g) + 1$, then $\langle I, t_1 g + t_2^d \rangle^{\mathbf{u}} = \langle I \rangle^{\mathbf{w}} (1 - \lambda^{\partial_{\mathbf{w}}(g)+1})$

Proof. The proof of *a)* and *b)* is immediate. For *c)*, choose a $\mathbf{w}$ compatible term-ordering τ on $R[t_1, t_2]$ for which $t_2 > t_1 > x_1 > \cdots > x_n$. Hence, by the rules in 1.117 and remarking that $\langle I \rangle^{\mathbf{w}} = \langle I \rangle^{\mathbf{u}}$

$$\begin{aligned}\langle I, t_1 g + t_2^{\partial_{\mathbf{w}}(g)+1} \rangle^{\mathbf{u}} &= \langle I, t_2^{\partial_{\mathbf{w}}(g)+1} \rangle^{\mathbf{w}} \\ &= \langle I \rangle^{\mathbf{w}} - \lambda^{\partial_{\mathbf{w}}(g)+1} \langle I : (t_2)^{\partial_{\mathbf{w}}(g)+1} \rangle^{\mathbf{w}} \\ &= \langle I \rangle^{\mathbf{w}} (1 - \lambda^{\partial_{\mathbf{w}}(g)+1})\end{aligned}$$

Remark 4.38 *Using the algorithm of proposition 4.37 we compute a more complex Gröbner basis with the Hilbert-Driven algorithms. More experiments are needed to determine if this method is more efficient than the straightforward use of proposition 4.36* **a.**

4.6.1 Non-principal saturations

In the general case, there are two main methods to compute a saturation:

1. the repeated transporter method, like in the principal saturation situation;
2. the reduction to a principal saturation through some "generic" element.

Since the possible combinations of algorithms for the transporter and saturation (especially in the non-principal case) are many, while almost every algorithm requires a (albeit slight) modification in the code of the Buchberger algorithm, we are still working on these algorithms. Our conjecture is that the repeated intersection/repeated transporter algorithms in their Hilbert-Driven versions are the best, since usually it pays to split the computations, but further experiments are needed.

4.7 Inverse images, multiple intersections, etc.

In this section we briefly show that the ideas above can be adapted to build algorithms to compute inverse images, multiple intersections and intersection of inverse images. These algorithms always allow a double interpretation, through module tag variable algorithms and through syzygies; we leave to the reader to check the details of the substantial identity of the two approachs. In these cases it is possible to use Hilbert function techniques as in the previous cases.

Inverse images

Let $R := k[x_1, \ldots, x_n]$, M be a free R-module, N a submodule of M generated by the elements $\{g_1 \ldots, g_m\}$, and $\phi : R \to M$ a module homomorphism determined by an element $f \in R$.
We want to find the inverse image $\phi^{-1}(N)$.

Remark 4.39 *It is easy to see that we obtain the result by eliminating the module variable t from $(tf - 1, tg_i)$. This result corresponds to intersecting the submodule generated by f with N, representing its elements in the form af, and taking a.*

Remark 4.40 *It is easy to see that $\phi^{-1}(N) = N :_R (f)$. Hence inverse images can be computed through the algorithms for transporter described later.*

In the same way, if

$$\begin{array}{cccc}\phi : & R^n & \longrightarrow & M \\ & \mathbf{e}_i & \rightsquigarrow & f_i\end{array}$$

is a module homomorphism, one obtains the inverse image $\phi_{-1}(N)$ eliminating the first component from $f_i + \mathbf{e}_i, g_j$ in $M \oplus R^n$.

Inverse images can be computed through syzygies: to compute an inverse image we have to combine an intersection (of the image with the module that we have to lift), a kernel, and add the kernel to any lifting of the elements of the image. Indeed, looking at the algorithm one sees that no forward-and-back is necessary, but just modifying the syzygy intersection algorithm slightly the inverse image is automatically computed. See for example [CT] if you want more details. The two algorithms are, as we have learned to expect, substantially identical.

Multiple intersections

If we have to intersect $N_1, \ldots, N_r \subseteq M$, we can add tag variables t_i, and eliminate them from

$$\sum_i t_i N_i + (1 + \sum_i t_i) R.$$

The t_i are either ring or module variables, the second possibility being more efficient.

Another algorithm can be made as follows: the intersection of many submodules $M_1, \ldots, M_r$ of M can be seen as the intersection of $\Delta \subseteq M^r$ with $M_1 \oplus \cdots \oplus M_r$, where Δ, the diagonal, is the image of the map

$$\begin{array}{lcl} M & @>> \quad > & M^r \\ m & \rightsquigarrow & (m, \ldots, m). \end{array}$$

We can compute the intersection with one of the algorithms for intersection of two submodules. The two algorithms sketched here substantially coincide. If it is more efficient using the algorithms described above or repeatedly apply the usual intersection algorithm depends on the input.

Intersection of inverse images

The intersection of many inverse images of N with respect to several maps $\phi_1, \ldots, \phi_r : R^n \to M$ is the inverse image of N^r with respect to the map $(\phi_1, \ldots, \phi_r)$ from R^n to M^r. This problem too can hence be solved with one Buchberger algorithm, whose details are left to the reader. Remark that the problem of intersection of many inverse images is the key to the algorithm describing the conductor of an integral extension described in [CT]; with these remarks the description of that algorithm (but experimentally not the computation itself) is simplified.

Chapter 5

Ideal Decompositions

In this chapter we give a brief resumé of the problems connected with some ideal decomposition algorithms (strict, equidimensional and reduced) and we describe a fast and quickly implementable algorithm that computes such decompositions. While no attempt to a full discussion of the problem has been done, the discussion of a more complex (surely non-basic) algorithm allows us to show situations where the necessity of the optimizations described in the previous chapters arise naturally. Moreover, the algorithm described here seems to be at least as efficient as the direct algorithms of [EHV] (see the discussion in subsection 5.5).

This chapter is organized in the following way:

a brief overview of the decomposition problems, section 5.1;

a description of the splitting techniques, section 5.2;

a discussion on the use of flatness in the projection to an hypersurface, section 5.3;

a description of the algorithm, section 5.4;

some statistics, section 5.5;

a detailed example, section 5.6.

5.1 A brief overview of the problem

The problem of decomposing an ideal into pure-dimensional components (if necessary reduced) is a key step in several basic algorithms of commutative algebra. The computation of the radical can be performed as intersection of the reduced components (but we wonder why you should really perform this intersection that throws away additional insight in the structure of the ideal), and the primary or the irreducible reduced decomposition can be more easily obtained from a pure-dimensional decomposition. Moreover this step can be performed independently of the ground field —at least if it is perfect— while the primary and irreducible decompositions require factorization, that depends on the ground field.

Several algorithms have been proposed for this computation, and fall mainly into two classes: the family of projection algorithms, whose prototype is the primary decomposition algorithm of [GTZ], and the direct, syzygy-based algorithms, like [EHV], see also [V].

The superiority of one type of algorithms over the other has not been settled; in [EHV] it is argued that the direct methods are superior since for projection methods "sufficiently generic" projections are needed. This assertion is only marginally true for the current literature — already in [AMR] "generic linear combinations" are needed only for finding a single univariate polynomial, through linear algebra and a variation of [FGLM], and the algorithms of [L] and [K2] compute triangular decompositions through characteristic sets without changes of coordinates. Moreover projection algorithms do not have the limitation to large characteristics of [EHV], in particular the example of D. Jaffe in [EHV] can be easily handled by our algorithm (and indeed by all projection algorithms).

In this paper we describe algorithms for equidimensional decompositions, that can be seen as a marginal modification of an algorithm of [AMR], but that completely avoid generic or random projections, and does not need lexicographic Gröbner bases or characteristic sets, hence can be a candidate to a best competitor against direct algorithms. Some tests seem to support the belief that our algorithms are much faster than direct algorithms.

Definition 5.1 *Let $R := k[x_1, \dots, x_n]$ and I be an ideal of R. Then the pure-dimensional ideals $I_1, \dots, I_k$ are*

1. *a* strict equidimensional decomposition *of I if $I = \cap I_j$;*
2. *an* iso-radical equidimensional decomposition *of I if $\cap\, I_j \supset I$ and $\sqrt{\cap I_j} = \sqrt{I}$;*
3. *a* reduced equidimensional decomposition *of I if each I_j is reduced and $\sqrt{I} = \cap I_j$.*

Remark 5.2 *The strict equidimensional decomposition is a step towards primary decomposition; the reduced equidimensional decomposition is a strong form of radical computation; and the iso-radical decomposition is simpler and sometimes sufficient, and already exhibits the main features. Moreover it is possible to recover the primary decomposition from a strict decomposition, as shown in [SY] and [K3].*

The basic computational tools used here are Gröbner bases with respect to suitable orderings, and ideal saturation, e. g. affine variety difference. We stress that no projection in generic (or random) direction, addition of random linear forms, etc. are used. The computation of the dimension and of the multiplicity are used to guide some steps of the algorithms.

The relation of flatness with variation of staircases, that is our main proof tool, has already been considered in [BGS]; specialization of Gröbner bases is already a main tool in [GTZ], see also [G] and [K1].

5.2 Splitting Techniques

The basic technique that we use to decompose an ideal $I \subseteq A$ is to split it according to an element $a \in A$.

Definition 5.3 *Let $R := k[x_1, \dots, x_n]$, I be an ideal of R and $a \in R$. Then the ideals $I_1, \dots, I_k \subset R$ are a* splitting *of I if $I_1 + \dots + I_k \supset I$ and $\sqrt{I_1 \cap \dots \cap I_k} = \sqrt{I}$. A splitting is* strict *if $I_1 \cap \ldots \cap I_k = I$. A splitting is* proper *if $I_j \supsetneq I$ for each $j : 1 \ldots k$.*

The basic operation on ideals that we use to split an ideal is the saturation, e. g. the closure of the set difference of the locus of I and the locus of J. This is accomplished through the ideal saturation. As seen in chapter 4, saturation computations are simpler if the ideal J is a principal. The basic tool to obtain a strict splitting is the following lemma, that is already the key step of [GTZ].

Proposition 5.4 *Let I be an ideal of the ring $R := k[x_1, \dots, x_n]$, a be an element of R and assume that $I : (a)^\infty = I : (a^k)$; then*

a) $I = (I : (a^k)) \cap (I + (a^k))$.

b) $\sqrt{I} = \sqrt{I : (a^k)} \cap \sqrt{I : (I : (a^k))^\infty}$.

Proof. The proof of *a)* is is quite easy: if $b = x + a^k y \in I$ with $x \in I$, $y \in R$, and $ba^k \in I$, then $a^{2k}y \in I$, hence $a^k y \in I$, and $b \in I$. The proof of *b)* follows from the remark: if we compare a primary decomposition of I with primary decompositions of $I : (a^k)$ and (I, a^k), in $I : (a^n)$ appear those associated primes of I that do not contain a, and in (I, a) appear primes associated to I that contain a, and possibly other components embedded in components of $(I : (a^n))$. □

Remark 5.5 *The decomposition in* a) *is strict, and moreover* $\sqrt{I} = \sqrt{I : (a^k)} \cap \sqrt{(I, a^k)}$. *The decomposition in* b) *is only iso-radical.*

Remark 5.6 *If the computation of the saturation is done by repeatedly use of the module tag variable algorithm, the computation of* $I : (a)^\infty$ *gives as a side effect the computation of the Gröbner basis of* (I, a^k) *for a suitable* k*, see 4.5.3.*

The advantage of case *b)* decompositions is the avoidance of the extraneous components introduced in (I, a^k) but in so doing we risk to discard some of the embedded components of I. It is not clear moreover if it is practically more convenient to avoid the extraneous components, or to avoid the more complicated saturation $I : (I : a^k)^\infty$; more experiments are needed to decide the issue.

The following well known result (see e. g. [AM]), is useful for the generalization of proposition 5.4 to non principal quotients.

Lemma 5.7 *Let* $R := k[x_1, \ldots, x_n]$ *and* $\wp$ *be a prime ideal,* $\mathfrak{q}$ *a* $\wp$*-primary ideal. Then the ideal* $\mathfrak{q} : (a)$

1. *is equal to* $\mathfrak{q}$ *if* $a \notin \wp$
2. *is equal to* $\wp$*-primary if* $a \notin \mathfrak{q}$
3. *is equal to* R *if* $a \in \mathfrak{q}$

Proposition 5.8 *Let* $R := k[x_1, \ldots, x_n]$ *and* I, J *be ideals of* R*. Then*

a) $\sqrt{I} = \sqrt{I : J^\infty} \cap \sqrt{(I, J)}$

b) $\sqrt{I} = \sqrt{I : J^\infty} \cap \sqrt{(I : J^\infty)^\infty}$

Proof. Case *a)* holds since it is true that if $\mathfrak{q}$ is $\wp$-primary and $J \not\subseteq \mathfrak{q}$ then $\mathfrak{q} : J^\infty$ is $\wp$-primary: it is intersection of $\wp$-primary ideals because of Lemma 5.7 and because $I : (J + K) = (I : J) \cap (I : K)$. Case *b)* follows from *a)* by the same reasoning of proposition 5.4. □

Other, simpler splittings are available

Proposition 5.9 *Let* $R := k[x_1, \ldots, x_n]$*,* I *be an ideal of* R *and* $a, b \in R$*. we have that :*

a) if $ab \in I$ *then* $\sqrt{(I, ab)} = \sqrt{(I, a)} \cap \sqrt{(I, b)}$.

b) If $a^k \in I$*, then* $\sqrt{(I, a^k)} = \sqrt{(I, a)}$.

Proof. For case *a)*, since $ab \in I$, we have that $I^2 \subset (I, a)(I, b) \subset I \Longrightarrow \sqrt{(I, a) \cdot (I, b)} = \sqrt{I}$. Thus $\sqrt{(I, a) \cap (I, b)} = \sqrt{(I, a) \cdot (I, b)} = \sqrt{I}$. Case *b)* follows immediately from case *a))*. □

Remark 5.10 *For strict splittings, we have to assure the properness of the splitting. I. e., a factoring splitting with* $a \in I$ *is not proper. But principal splittings may be non proper even if* $a \notin I$ *and* $b \notin I$*. If* $a \in \sqrt{I}$*, then* $I : (a)^\infty = (1)$*, and* $(I, a^k) = I$ *for sufficiently large* k*, hence the splitting is not proper. Hence, for strict splittings, we use only principal splittings with splitting elements not in the radical.*

A summary of the splittings we have introduced:

Principal: $I \to (I : (a), (I, a))$ with $a \in R$ and $I : (a) = I : (a)^\infty$;

Loose: $I \to (I : J^\infty, I + J)$

Bi-saturating: $I \to (I : J^\infty, I : (I : J^\infty)^\infty)$;

Factoring: $I \to ((I, a), (I, b))$ when $ab \in I$

Radical: $I \to (I, a)$ when $a^n \in I$

5.3 Staircases, flatness and hypersurfaces

5.3.1 Families of Gröbner Bases and block orderings

Definition 5.11 *Let $R := k[x_1, \dots, x_n]$, $\tau \in \mathrm{TO}(R)$ and I be an ideal of R. Then the* staircase *of I is the set of the terms in T^n that don't belong to $\mathrm{Lt}_\tau(I)$. It is a vector space basis of $k[x_1, \dots, x_n]/I$.*

We have seen in chapter 1 that the computation of the dimension (hence codimension) and multiplicity of an ideal may be done through its Hilbert function. Another algorithm, [BW] computes the codimension through the computation of the complementary set of a maximal subset of algebraically independent variables. We may hence suppose, up to a reordering, that $\dim(R/I) = c$ and that the variables $x_1, \dots, x_c$ are algebraically independent. Let $R := k[x_1, \dots, x_n]$, and consider a partition of the variables of R into two subsets $\mathbf{x}$ and $\mathbf{y}$. Let τ be an elimination term-ordering for $\mathbf{x}$.

Considering the ordering τ the ring $k[\mathbf{x}, \mathbf{y}]$ can be naturally seen as $(k[\mathbf{y}])[\mathbf{x}]$, e. g. the set of the polynomials in $\mathbf{x}$ with coefficients in $k[\mathbf{y}]$. If I is an ideal of $k[\mathbf{x}, \mathbf{y}]$, and $I_1 = I \cap k[\mathbf{y}]$, we can see $k[\mathbf{x}, \mathbf{y}]/I = (k[\mathbf{y}]/I_1)[\mathbf{x}]/(I/I_1)$. If in particular I_1 is a prime ideal, we can see the polynomials of $k[\mathbf{x}, \mathbf{y}]$ as polynomials with coefficients in the quotient field $Q(k[\mathbf{y}]/I_1)$ of $k[\mathbf{y}]/I_1$; hence the question arises if the image of a reduced Gröbner basis is still a Gröbner basis, and the answer is affirmative. Consider the map

$$\phi : k[\mathbf{x}, \mathbf{y}] \to k[\mathbf{x}, \mathbf{y}]/I_1 = k[\mathbf{x}, \mathbf{y}] \otimes_{k[\mathbf{y}]} k[\mathbf{y}]/I_1 \to k[\mathbf{x}, \mathbf{y}] \otimes_{k[\mathbf{y}]} k[\mathbf{y}]/I_1 = \mathfrak{q}(k[\mathbf{y}]/I_1)[\mathbf{x}]$$

this means considering $f \in k[\mathbf{x}, \mathbf{y}]$ as a polynomial in $\mathbf{x}$ with coefficients in $k[\mathbf{y}]$ that are considered mod I_1. Hence, we may write a polynomial $f \in k[\mathbf{x}, \mathbf{y}]$, $f = c\mathbf{x}^{\boldsymbol{\alpha}}\mathbf{y}^{\boldsymbol{\beta}} + \dots$ as a polynomial in $(k[\mathbf{y}])[\mathbf{x}]$, e. g. $f = \mathbf{x}^{\boldsymbol{\alpha}} g(\mathbf{y})$, and $\mathrm{Lt}_\tau(g) = \mathbf{y}^{\boldsymbol{\beta}}$.

Theorem 5.12 *Let $R := k[x_1, \dots, x_n]$, $\{\mathbf{x}, \mathbf{y}\}$ be a partition of $\{x_1, \dots, x_n\}$, $\tau := \sigma_{\mathbf{x}} \times \sigma_{\mathbf{y}}$ be an elimination term-ordering for $\mathbf{x}$ and let I be an ideal such that $I_1 = I \cap k[\mathbf{y}]$ is prime; let G be a Gröbner basis of I. Then the image of G in $(k[\mathbf{y}]/I_1)[\mathbf{x}]$ is a Gröbner basis with respect to $\sigma_{\mathbf{x}}$.*

Proof. We may assume that G is irrendundant. This means that if we have $f \in G$, $f = \mathbf{x}^{\boldsymbol{\alpha}} g(\mathbf{y})$, $\boldsymbol{\alpha} \neq 0$, we have $g \notin I_1$, otherwise g would be reducible by an element of I_1, hence by another element of G; this implies that for elements of G if $\mathrm{Lt}_\tau(f) = \mathbf{x}^{\boldsymbol{\alpha}}\mathbf{y}^{\boldsymbol{\beta}}$, then $\mathrm{Lt}_\tau(\phi(f)) = \mathbf{x}^{\boldsymbol{\alpha}}$. Recall that a set G of elements of an ideal is a Gröbner basis if and only if every syzygy between the initial power product of two elements of G can be extended to a syzygy between the elements of G. Extending a syzygy means to find a syzygy beginning with the same monomials in the standard syzygy ordering induced by the ordering of the current ring. (This is just a rephrasing of the criterion that a set of generators is a Gröbner basis if and only if every S-polynomial is reducible to 0). Let now $\bar{\sigma}$ be a syzygy between $\mathrm{Lt}_\tau(\phi(f_i))$ and $\mathrm{Lt}_\tau(\phi(f_j))$, $f_i, f_j \in G$; this identifies a syzygy σ between $\mathrm{Lt}_\tau(f_i)$ and $\mathrm{Lt}_\tau(f_j)$ obtained multiplying by suitable powers of some $\mathbf{y}$ variables. This syzygy can be lifted to a syzygy Σ of G, since G is a Gröbner basis, and identifies a syzygy $\bar{\Sigma}$ of $\phi(G)$ that lifts $\bar{\sigma}$. This proves that $\phi(G)$ is a Gröbner basis. □

Remark 5.13 *Remark that in the proof the key point is that $\mathrm{Lt}_\tau(f)$ and $\mathrm{Lt}_\tau(\phi(f))$ only differ by the power product in the $\mathbf{y}$ variables; One can remark that this is not necessary to be true for all the elements of G, but that before checking it, one can delete redundancies in $\phi(G)$.*

Definition 5.14 *Let A be a ring. Then a* geometric point *y of A is a field that is an A-module, and if B is another A-module and $f \in B$, then $f(y)$ is $f \otimes 1$ in $B \otimes_A y$. In particular*

1. *a prime ideal $\wp$ of A (e. g. an element of $Spec(A)$) defines a geometric point being the quotient field of $A/\wp$;*
2. *if A is a domain, the 0 ideal identifies a geometric point (the generic point), and Theorem 5.12 says that if I_1 is prime then the image of a Gröbner basis at the generic point is still a Gröbner basis.*

Definition 5.15 *Let $R := k[x_1, \dots, x_n]$, $\mathbf{x}, \mathbf{y}$ a partition of $\{x_1, \dots, x_n\}$, τ a elimination term-ordering for the $\mathbf{x}$ variables and $f \in R$. Then the* reduced initial term *$\mathrm{Lt}_{\tau,\mathbf{x}}(f)$ is the term $\mathrm{Lt}_\tau(f)$ from which the variables $\mathbf{y}$ are deleted. Expressing f as polynomial in the $\mathbf{x}$ variables with coefficients in the $\mathbf{y}$ variables, we have $f = \mathrm{Lt}_{\tau,\mathbf{x}}(f)g(\mathbf{y}) + h$, all the monomials of h being smaller than $\mathrm{Lt}_{\tau,\mathbf{x}}(f)$. We denote $\mathrm{lc}_{\tau,\mathbf{x}}(f) = g(\mathbf{y})$.*

Theorem 5.16 *With the notation of theorem 5.12, for almost all (in the sense of Zariski topology) prime ideals $J \in \mathrm{Spec}(k[\mathbf{y}]/I_1)$ the image of G in $(k[\mathbf{y}]/J)[\mathbf{x}]$ is a Gröbner basis with respect to $\sigma_{\mathbf{x}}$.*

Proof. We can put the thesis of Theorem 5.12 as a question for the geometric points of $Spec(k[\mathbf{y}]/I_1)$, e. g. consider $J \in \mathrm{Spec}(k[\mathbf{y}]/I_1$, instead of I_1, and $I + J$ instead of I. The answer is that for an open subset of $Spec(k[\mathbf{y}]/I_1)$ the Gröbner basis of I is still a Gröbner basis (usually redundant). If I_1 is prime this set is non-empty since it contains the generic point. It is computed as follows: Let G be the reduced Gröbner basis of I with respect to $\tau \in \mathrm{TO}(R)$. Extract from G a minimal set G' of polynomials, not in I_1, such that every other element f of G not in I_1 is such that $\mathrm{Lt}_{\tau,\mathbf{x}}(f)$ is multiple of $\mathrm{Lt}_{\tau,\mathbf{x}}(f')$ for a suitable element $f' \in G'$. The set G' may be not unique. For each element of $h_i \in G'$ consider $h_i = g_i \,\mathrm{Lt}_{\tau,\mathbf{x}}(h_i) + h'_i$, where $g_i \in k[\mathbf{y}]$ and the power products of h'_i are smaller than $\mathrm{Lt}_{\tau,\mathbf{x}}(h_i)$. The h_i are non-zero in $k[\mathbf{y}]/I_1$ since they are reduced modulo the Gröbner basis of I_1. Remark that, because of the reduction hypothesis on G, no g_i is in I (otherwise one could reduce g_i through elements of $G \setminus G'$). Hence, for each geometric point y of I_1 for which no $g_i(y)$ vanishes, we have that $G'_y = \{f(y) \mid f \in G'\}$ is a Gröbner basis of y with respect to $\sigma_{\mathbf{x}}$ with the same proof given for theorem 5.12. Thus, if I_1 is prime, since each of the g_i does not vanish on the locus of I_1, there is a dense open subset of $Spec(k[\mathbf{y}]/I_1)$ where no g_i vanishes, and in this open set G specializes to a Gröbner basis. □

Remark 5.17 *The image of G may be built using the algorithm sketched in the proof of theorem 5.16.*

Remark 5.18 *Usually we don't know if I_1 is prime, but we may check if the product $\prod g_i$ is in $\sqrt{I_1}$. If this is the case, we have discovered that I_1 is not prime, and we can split I_1 (and I) into smaller components. Otherwise $\prod g_i$ does not vanish on a non-empty open subset of $Spec(k[\mathbf{y}]/I_1)$ even without the primality assumption. In this case we have found a non-empty open subset such that on it a Gröbner basis specializes.*

Remark 5.19 *Remark that every different choice of G' gives a (possibly different) open subset where the staircase is constant; and it is even possible that the Gröbner basis specializes to a Gröbner basis with a different staircase.*

The situation is more complicated when I_1 is not reduced, since it may happen that some g_i vanishes identically, and we might not be allowed to split I_1 since we are looking for strict splittings. The situation will be further discussed.

5.3.2 Staircase variation and flatness

Consider this situation: we have an ideal $I \subseteq k[\mathbf{x}, \mathbf{y}]$ and $J = I \cap k[\mathbf{y}]$; consider the projection morphism between the corresponding algebraic varieties, induced by the map

$$k[\mathbf{y}]/J \to k[\mathbf{x}, \mathbf{y}]/I$$

We want to find an open subset where the map is flat. We have seen in the previous section how to find an open subset where the staircase is constant, and either this subset is non empty or we find a splitting (except a special case that will be discussed separately).

Lemma 5.20 *Let $R := k[x_1, \dots, x_n]$ and I, J be ideals of R. Then if we have a map*

$$Spec(k[\mathbf{x}, \mathbf{y}]/I) \to Spec(k[\mathbf{y}]/J)$$

and the Gröbner basis of the fiber has constant staircase in an open subset of $Spec(k[\mathbf{x}, \mathbf{y}]/I)$, then the map is flat on the same subset.

Proof. This can be easily seen e.g. if the generic fiber is finite (e. g. the staircase is closed) since flatness is implied by finiteness and constant k-dimension of the fiber; and this dimension is computed through the staircase (it is the multiplicity). In the higher dimensional case, one goes through homogeneization (with respect to a suitable weight, such that the Gröbner basis is still Gröbner with respect to an ordering compatible with the weight) and considering the filtration induced by the grading; then the Hilbert function is constant, and this proves the flatness (see e.g. [Mu], or [E]). □

We want to be more precise, generalizing the result that the staircase freely generates the quotient. As in the previous section, let $R := k[x_1, \dots, x_n]$, $\mathbf{x}, \mathbf{y}$ a partition of $\{x_1, \dots, x_n\}$, $\tau := \sigma_{\mathbf{x}} \times \sigma_{\mathbf{y}}$ an elimination term-ordering for $\mathbf{x}$, let I be an ideal such that $I_1 = I \cap k[\mathbf{y}]$ is prime; let G be a Gröbner basis of I; then the image of G in $(k[\mathbf{y}]/I_1)[\mathbf{x}]$ is a Gröbner basis of I_1 with respect to $\sigma_{\mathbf{y}}$. G be the reduced Gröbner basis of an ideal $I \subseteq k[\mathbf{x}, \mathbf{y}]$; extract from G a minimal set $\{h_i\} = H$ of polynomials, not in $I_1 = I \cap k[\mathbf{y}]$, such that every other element f of G not in I_1 is such that $\mathrm{Lt}_{\tau,\mathbf{x}}(f)$ is multiple of $\mathrm{Lt}_{\tau,\mathbf{x}}(h_i)$ for a suitable element h_i, and let $h_i = g_i \,\mathrm{Lt}_{\tau,\mathbf{x}}(h_i) + h'_i$. Let E be the open subset of $Spec(k[\mathbf{y}]$ where the g_i do not vanish (it may be empty).
Consider the $\mathbf{x}$-*staircase* S of I, being the set of $\mathbf{x}$-power products that are not multiple of some $\mathrm{Lt}_{\tau,\mathbf{x}}(f)$, $f \in I$ (or, equivalently, $f \in G$). We have seen that S is the staircase of $I(\mathbf{y})$ for every geometric point $\mathbf{y}$ of E.

Theorem 5.21 *Let E be an R module. If E is not empty, then the $\mathbf{x}$-staircase S is a set of free generators of $k[\mathbf{x}, \mathbf{y}]_E$ as $k[\mathbf{y}]_E$-module.*

Proof. On E the g_i are invertible, and the h_i/g_i are monic in $\mathbf{x}$, hence S generates $k[\mathbf{x}, \mathbf{y}]_E$; and S is linearly independent on $k[\mathbf{y}]/I_1$ since G is a Gröbner basis, hence linearly independent on $(k[\mathbf{y}]/I_1)_E$ by the flatness of localization. □

Remark 5.22 *This proves that $k[\mathbf{x}, \mathbf{y}]/I$ is $k[\mathbf{y}]/I_1$-flat on E; changing E through a different choice of H (or even through different term-orderings) one can find other open subsets where $k[\mathbf{x}, \mathbf{y}]/I$ is flat.*

The flatness is important because of this lemma:

Lemma 5.23 *Let A be a pure-dimensional ring, and B a finite flat A-algebra. Then B is pure-dimensional.*

Proof. Consider a prime $\mathfrak{q}$ of B associated to (0), and let $b \in B$ be an element annihilated by $\mathfrak{q}$; because of flatness, $\mathfrak{q} \cap A$ is a prime of A associated to (0), and because of the finiteness hypothesis the dimension of $\mathfrak{q}$ is equal to the dimension of $\mathfrak{q} \cap A$. □

5.3.3 Projecting to an hypersurface

Consider an ideal $I \subseteq k[x_1, \ldots, x_n]$, and the corresponding affine variety $V := \mathbf{V}(I)$; assume that V has dimension d. We want to project V into a subspace of dimension $d+1$. We have seen in chapter 4 that this is possible through a Gröbner basis computation with respect to an elimination term-ordering. We don't want to use a "sufficiently generic" projection like [AMR], because we want to keep the problem as sparse as possible. We use a very special projection, a projection along coordinates on a hypersurface.

We have compute the dimension of V. First, we compute a Gröbner basis with respect to an ordering whatsoever, then we use the recursive algorithm for the computation of the dimension 1.4.4. This algorithm gives as a side effect the identification of a maximal subset of algebraically independent variable in $\{x_1, \ldots, x_n\}$. We then choose a subset $\mathbf{y}$ of $d+1$ variables that contain an algebraically independent set. To ease the notation, we assume that these are the last $d+1$ variables. Let $c = n - d$ be the codimension, we have that $\mathbf{x} = (x_1, \ldots, x_{c-1})$ and $\mathbf{y} = (x_c, \ldots, x_n)$. Let τ be an elimination term-ordering for the first $c-1$ variables. We compute the Gröbner basis of I with respect to τ, and the elimination ideal $I' = I \cap k[\mathbf{y}]$. The ideal I' describes an affine variety V' that is the projection of V along the coordinates $x_1, \ldots, x_{c-1}$ If the ideal I is in "general position", then the map from V to V' is generically 1-1, irreducible components of V and V' are in 1-1 correspondence, the dimension of the components is preserved, and it is impossible that V' is reduced if V is not reduced; hence in particular V is pure-dimensional and reduced if and only if V' is pure dimensional and reduced, e. g. I' is generated by a squarefree polynomial g. If not, consider the squarefree GCD g of a system of generators of I'; we can add g to I to discard the lower-dimensional components and at least a part of nilpotents.

With a special projection, like the ones to which we are bound, several phenomena can happen:

1. some components of V may be projected into smaller dimension,
2. some components of V may disappear inside other components,
3. the map may be of degree > 1 on some components,
4. some embedded components may disappear altogether,
5. some non-reduced components may have reduced image.

We can take care of all these phenomena through flatness, except the last one, that is important only for the reduced decomposition.

5.4 The algorithms

Now we describe the algorithms. We have three variants, for strict, iso-radical and reduced equidimensional decomposition. The algorithm for iso-radical decomposition can be seen as the main one; it has some optional steps, that are probably useful for the iso-radical decomposition, are necessary for the reduced decomposition, and forbidden for the strict decomposition. Moreover, for the strict decomposition all the splittings have to be strict, for the other decompositions we can choose other splittings. And for the reduced decomposition there is a final step that will be described separately.

We first describe the algorithm for the iso-radical decomposition. The algorithm for the reduced decomposition 5.4.3 is easily derived from that. the algorithm for strict decomposition 5.4.4 requires some more caution in the splittings. Since the algorithm are quite complex, there is no pseudo-code, but only a step-by-step description.

5.4.1 Iso-radical decomposition algorithm

The algorithm proceeds inductively, finding for every ideal I either a proof that it is equidimensional or a proper splitting into two ideals (a special case being the addition to an ideal of a root of one element). The algorithm terminates by noetherianity; but indeed one can also remark that in every splitting that we consider either the dimension or the multiplicity or the number of associated primes decreases for each factor.

We describe the algorithm in steps, in which a splitting might be discovered (and hence we have to repeat from the first step for each of the components). At the end of all the steps we have found one component of the sought decomposition. When we have all the components, we still have (optionally) to eliminate redundancies.

Step 1 *Compute the dimension d of I, $c = n - d$, and if needed reorder the variables in such a way that the last d are algebraically independent mod I. Compute a Gröbner basis with respect to an elimination term-ordering for the first $c-1$ variables, thus computing $I' = I \cap k[\mathbf{y}]$. If I' is not a principal ideal, then let g be the GCD of its generators, and split I along g; otherwise go to the next step.*

Step 2 *If g is square-free, go to the next step; otherwise split g through a square-free decomposition, and split I accordingly. This step is optional, but usually it speeds the algorithm.*

Let G be the reduced Gröbner basis of I (with respect to the current block ordering), and erase g from G; extract from G a minimal set G' of polynomials such that given $h \in G$ exists $h' \in G'$, and $\mathrm{Lt}_{\tau,\mathbf{x}}(h')$ divides $\mathrm{Lt}_{\tau,\mathbf{x}}(h)$ (the set G' may be not unique). For each element of $h_i \in G'$ consider $h_i = g_i \, \mathrm{Lt}_{\tau,\mathbf{x}}(h_i) + h_i'$, where $g_i \in k[\mathbf{y}]$ and the power products of h_i' are smaller than $\mathrm{Lt}_{\tau,\mathbf{x}}(h_i)$. Clearly no g_i is multiple of g, since every h_i is in normal form with respect to g.

Step 3 *If some g_i has a common factor with g, split I along g_i.*

We assume now that no g_i has a factor in common with g. Define $g' = \prod g_i$. Then g' is not in I.

Step 4 *Compute $I'' = I : (g')^\infty$; if $I'' \neq I$ split along g'.*

Now the map $\mathrm{Spec}(k[\mathbf{x},\mathbf{y}]/I) \to \mathrm{Spec}(k[\mathbf{y}]/(g))$ is flat where g' does not vanish; this means that in these points the dimension is d, and since I is saturated with respect to g' this means that no associated prime contains g'; hence I is of pure dimension d.

Step 5 *Conclusion: I is pure-dimensional*

Remark 5.24 *Some variants :*

a) An alternative for step 4 is to split for each of the g_i separately; this is probably more convenient, since each transporter is easier, and it is usually a good technique to have several easier steps instead of an harder one.

b) Another alternative consists in avoiding the arbitrary choice of the h_i; more precisely, erase from $G' = G \setminus \{g\}$ all the elements h such that $\mathrm{Lt}_{\tau,\mathbf{x}}(h)$ is a proper multiple of $\mathrm{Lt}_{\tau,\mathbf{x}}(h')$ for another $h' \in G'$; then partition G' into classes H_i of elements having the same $\mathrm{Lt}_{\tau,\mathbf{x}}$, and for each class consider the ideal J_i generated by the $C_{\mathbf{x}}(h)$, $h \in H_i$. Then we can use loose or bi-saturating splittings with the J_i.

c) A further possibility is to try to factor all the elements of G, and split with every possible factor, either with factoring splittings or with strict splittings if needed.

5.4.2 Optimizations

One of the main advantages of the projection methods, and hence of our algorithm, is the possibility of discover, during the computation, useless or trivial branches. Trivial, but useful, cases arise if the ideal is linear or monomial. This is not infrequent after some splitting.

Remark 5.25 *Suppose that we know an upper bound on the codimension of the associated primes, for example from the number of generators of the input ideal. Then in the algorithm we may discard all the ideals that are of codimension higher than the limit, since it is sure that they will be redundant in the final decomposition.*

Remark 5.26 *If the ideal is zero-dimensional, it is obviously pure-dimensional.*

Such cases occur very often throughout the branching process, especially on the last recursive steps. Other optimizations are substantial : the Hilbert-Driven algorithm, a factoring Buchberger algorithm. Using the saturation algorithms described in chapter 4, the computation of (I, a^k) is a side-effect of the computation of $(I : a^k) = I : a^\infty$.

When the input ideal is homogeneous with respect to a system of weights $\mathbf{w}$, when we compute the first Gröbner basis to have the dimension, we have easily the Hilbert function too. The Hilbert-Driven algorithm may hence be used en lieu of the usual Buchberger algorithm, and the Hilbert-Driven version of the saturation algorithm may be used too.

If it is available a factoring Buchberger algorithm, see chapter 2, every Gröbner basis computation may be performed with it. Every branch of the factoring algorithm is then treated as a split. Since the number of branches may be very high, some care has to be taken in using the factoring method.

5.4.3 Reduced decomposition algorithm

If we are looking for a reduced decomposition, we have not yet finished the work, since I might still have nilpotents.

Some trivial cases appear frequently among the ideals produced by the iso-radical decomposition algorithm :

Lemma 5.27 *Let $R := k[x_1, \ldots, x_n]$ and I be an ideal of R. Then*

1. *If I is linear, then I is radical.*
2. *If $I := (g)$ then $\sqrt{I} = (\mathrm{SQFR}(g))$.*
3. *If $I = (t_1, \ldots, t_k)$ is monomial, then $\sqrt{I} = (\mathrm{SQFR}(t_1), \ldots, \mathrm{SQFR}(t_k))$.*

Some less trivial cases are the following

Remark 5.28 *Let $R := k[x_1, \ldots, x_n]$ and I be an equi-dimensional ideal of R. Then if the multiplicity of I is equal to the multiplicity (the degree) of g, then the ideal is radical.*

Proposition 5.29 *Let $I \subset k[x_1, \ldots, x_n]$ be an ideal, and $\mathrm{Lt}_\tau(I)$ its leading term ideal of I with respect to some term-ordering τ. Then $Lt_\tau(I)$ radical $\Rightarrow$ I radical.*

Proof. Let $f \in R$ such that $f^k \in I$. Then

$$\left.\begin{array}{l}\mathrm{Lt}_\tau(f^k) \in \mathrm{Lt}_\tau(I)\\ \mathrm{Lt}_\tau(f^k) = \mathrm{Lt}_\tau(f)^k\end{array}\right\} \Longrightarrow \mathrm{Lt}_\tau(f)^k \in \mathrm{Lt}_\tau(I) \Longrightarrow \mathrm{Lt}_\tau(f) \in \mathrm{Lt}_\tau(I)$$

hence $\exists\, g \in I$ s. t. $\mathrm{Lt}_\tau(g) = \mathrm{Lt}_\tau(f)$, hence $(f-g)^k = f^k + g \cdot (\ldots) \in I$. But $\mathrm{Lt}_\tau((f-g)^k) \in \mathrm{Lt}_\tau(I) \Longrightarrow \mathrm{Lt}_\tau((f-g)) \in \mathrm{Lt}_\tau(I)$. Then, there exists $g_1 \in I$ s. t. $Lt_\tau(g_1) = \mathrm{Lt}_\tau(f-g)$ and the thesis follows by recursion. $\square$

These remarks allow to cheaply recognize some reduced ideal; indeed, in all cases the checks are quite easy if we have a Gröbner basis.
In general, more work is required.

To test if an equi-dimensional ideal is reduced, or to compute the radical of such an ideal, we may use the 0-dimensional test techniques of [AMR] or the jacobian techniques of [EHV]. Since the tests and the computation of the radical using these techniques may be quite expensive, we prefer to use a variant of the iso-radical algorithm. It is not sure that this is the best way, but at least it is not too bad:

We need the some trivial remarks:

Remark 5.30 *If $\phi : V \to W$ is an affine variety morphism, the Zariski tangent space of $\wp \in V$ is mapped into the Zariski tangent space of $\phi(\wp)$.*

Remark 5.31 *If in the lemma above ϕ is induced by a projection, the Zariski tangent space of $\wp$ is contained in the sum of the tangent space at $\phi(\wp)$ plus the space parallel to the projection direction.*

Corollary 5.32 *If V embedded in an affine space of pure dimension d has a singular point $\wp$, then exists a projection π along a coordinate space into an affine space of dimension $d+1$ such that $\pi\wp$ is singular.*

Indeed, the Zariski tangent space, that has dimension $> d$, cannot project non-injectively in every direction.

With these remarks we can try different directions, and either find one that allows us to proceed inductively, or prove that $\hat{V}$ is not only pure-dimensional, but also generically non-singular, hence reduced. The conclusion can be obtained when the intersection of the projection directions is empty. In some cases this can be a very expensive part; for example, if I is the ideal $(x_i^2 - x_i)^2_{i=1\ldots n}$ we need n projections to find the radical. But in this case also the other known algorithms are quite expensive. If I is of dimension d in a polynomial ring with n variables in general we need $m \geq n/(d+1)$ different projections, since every computation with a different choice of independent variables decreases of a dimension $d+1$ the directions in which nilpotents are concealed.

The idea of the reduced decomposition algorithm is : for every direction, compute the Gröbner basis of the input ideal I. If the resulting projection ideal I' is not principal, find a principal ideal (g) by recursion. Otherwise, take the squarefree h of g and, if h doesn't belong to the ideal, add it the the ideal and restart. After the examination all the directions, if no branching happened, the ideal is reduced.

Remark 5.33 *Let I be pure-dimensional, non-reduced ideal. If $a \in \sqrt{I} - I$, not necessarily (I, a) is pure-dimensional.*

Hence, after adding the squarefree element to the ideal, we have to use the iso-radical decomposition algorithm again to have a pure-dimensional decomposition.

Reduced decomposition algorithm

The input $I \subset R$ is pure-dimensional. The Gröbner basis for some term-ordering τ is already been computed by the iso-radical decomposition algorithm.

Step 1
Compute the dimension d of I, $c = n - d$.

- *If I is principal or monomial then return* $\mathrm{SQFR}(I)$.
- *If I is linear or* $\mathrm{Lt}_\tau(I)$ *is radical then return I.*

Compute the list O of all the necessary set of variables (those corresponding to the necessary directions). For each set $\mathcal{V}$ in L let $\mathcal{I}$ be the complementary set with respect to $\{x_1, \ldots, x_n\}$ and do the following steps :

Step 2
Let σ be an elimination term-ordering for $\mathcal{V}$. Then compute G_σ the Gröbner basis of I with respect to σ. If $\mathrm{Lt}_\tau(I)$ is radical, the return I.
Let $I' := I \cap k[\mathcal{I}]$. If I' on I'. Let $I'' := (g)$ be the resulting principal ideal.

Step 3
Let $g' := \mathrm{SQFR}(g)$. If $\deg(g') = \mathrm{mult}(I)$ the return I.

Step 4
If $g' \notin I$ then let L be the list of pure-dimensional ideals produced by the application of the iso-radical decomposition algorithm to the ideal (I, g'). Return the union of the result of the application of the reduced decomposition algorithm to every element of L.

Step 5
Conclusion: I is reduced. Return I.

In the case of reduced decomposition, it may be of interest to intersect the reduced equi-dimensional ideals produced by the algorithm, obtaining the radical. If the input ideal is homogeneous with respect to some system of weights, this can be done using the Hilbert-Driven intersection algorithm shown in chapter 4.

5.4.4 Strict decomposition algorithm

If we are looking for a strict decomposition, we are allowed to use only strict splittings. Some modification to the iso-radical algorithm are necessary :

a) Step 2 has to be skipped.

b) In every other step, use only strict splittings, see section 5.2.

c) In step 3, it may be that we cannot produce a strict splitting since g_i is in the radical of I_1, e. g. a power of g_i is a multiple of g. In this case $k[\mathbf{x}, \mathbf{y}]/I$ is not generically flat over $k[\mathbf{y}]/I_1$.

   Assume now that the first of the $\mathbf{y}$ (let it be y_1) is algebraically dependent from the others in $k[\mathbf{y}]/I_1$, e. g. that y_1 appears in g (one can always assume that, up to a permutation of the variables). Let $\mathbf{y}' = \mathbf{y} \setminus \{y_1\}$, $\mathbf{x}' = \mathbf{x} \cup \{y_1\}$ and change the ordering to a block ordering eliminating $\mathbf{x}'$. Recompute the Gröbner basis of I. Clearly $I \cap k[\mathbf{y}'] = (0)$.

   We now can proceed as in the main algorithm: find a polynomial g' in $k[\mathbf{y}']$ such that the map $k[\mathbf{x}, \mathbf{y}]/I \to k[\mathbf{y}']$ is flat outside of the locus of g', ($g' \notin I$ since $I \cap k[\mathbf{y}'] = (0)$) and continue with step 4.

Remark 5.34 *The use of* c) *for non-strict decomposition is counterproductive : it does not allow to discover and eliminate nilpotents, (hence it cannot be used for the reduced decomposition) and eliminating one further variable makes the computations more costly.*

5.5 Tests

A first complete experimental implementation of algorithm has been made in CoCoA, using the high-level language CoCoAL, (in particular the GCD is implemented through Gröbner basis computations). The implementation is much less efficient since it does not include all the heuristics and simplifications that are natural in a semi-automatic computation.

There is no established benchmark for decomposition and radical computations, hence we propose some of our tests, that are "natural examples", existing in the literature. All the examples are contained in the POSSO test suite.

G Gerdt example, see [BGK];

B Butcher example, see [BGK];

H2,H3 Two examples of Hietarinta, coming from theoretical physics problems; they are contained, as Hietarinta-1 and Hietarinta-3 in the POSSO test suite (the Hietarinta-2 problem is a trivial ideal).

8_3 The hypothesis of the Maclane 8_3 theorem: $(C+cE-eC-E,\ F-C,\ E-G,\ eF+fH+hE-fE-hF-eH,\ fG-gF,\ gH+G-hG-H,\ cH-hC)$: see [CT] where (substantially) the real radical of this ideal is computed.

E2,E3 Two examples of Amrhein-Gloor-Küchlin, see [AGK]; they are of dimension zero, hence are interesting only for the computation of the radical.

E2 $(x+3xy^3+y^4+yz^2,\ -x^2z+2y^3z+z^2+2yz^2+3xyz^2,\ 3x^3+xy^2+yz^2-2xz^3)$,
E3 $(x^2+y^4+x^3z+yz-2xz^3,\ -x^2y^2-y^3z-z^3-3yz^3,\ y^4-x^2z+2y^2z-2xyz^2)$.

L An example of Logar (personal communication): $(2ahi+bh^2+2cdj-cei-cgh-deh,\ ai^2+2bhi+2cfj-cgi+d^2j-dei-dgh-efh,\ bi^2+2dfj-dgi-efi-fgh,\ f(fj-gi))$.

O Oaku's G_3 (see [SY]): $(((12y+8)x^2+(2y+2)x)\eta\xi+((-15y^2-4y)x-4y^2-y)\eta^2, -x\xi^2+((-12y-8)x+2y)\eta\xi+(15y^2+4y)\eta^2, (81y^4x^2+(-54y^3-12y^2)x-12y^3-3y^2)\eta^3, (-24yx+6y^2-6y)\eta^2\xi+(-81y^4x+81y^3+24y^2)\eta^3, (48x^2+(-30y+12)x-6y)\eta^2\xi+((81y^3-54y^2-24y)x-21y^2-6y)\eta^3, (-96yx-18y^3+18y^2-24y)\eta^2\xi+(243y^5x-243y^4+72y^3+48y^2)\eta^3, 6y\eta^2\xi^2+((576y+384)x^2+(-81y^3-306y^2-168y+96)x+81y^2-18y)\eta^3\xi+((-720y^2-192y)x+450y^3-60y^2-48y)\eta^4)$.

M an example from the Macaulay2 tutorial: $(-a^3d+b^4,\ -a^3c+ab^3,\ -ac^3d+ad^4+bc^4-bcd^3,\ -bc^3d^2+bd^5+c^6-c^3d^3,\ ac^5-ac^2d^3-b^2c^3d+b^2d^4,\ -a^3d^3+a^2c^4-a^2cd^3+b^3d^3,\ -a^3d^3+b^3c^3,\ -a^3cd^2+ab^2c^3-ab^2d^3+b^3cd^2,\ -a^3c^2d+a^2bc^3-a^2bd^3+b^3c^2d,\ -a^3bd^2+a^3c^3,\ a^4c^2-a^3b^2d)$.

Here is a table with some results.

The computations have been performed using as coefficient field $\mathbb{Z}_{32003}$. The Hilbert-Driven and factoring Buchberger algorithms has not been used for the testing. The timings are only for the Gröbner bases and saturation computations. Since the algorithm is at the moment implemented in a prototypical CoCoAL version, the other (auxiliary) operations, done in high-level CoCoAL code, are quite inefficient. The final version will be implemented in Aldor using the PoSSo library, and the auxiliary operations will be efficiently implemented. We expect that the auxiliary operation will be negligible with respect to Gröbner basis computations and saturations.

	Dec	**Rad**	**Tot**
Gerdt	20.16 s	3.82 s	23.98 s (15.32 s)
Butcher	25.69 s	0.66 s	26.35 s (2.97 s)
H1	43.05 s	15.47 s	58.52 s
H3	5.52 s	3.48 s	9.00 s
8_3	6.23 s	20.42 s	26.65 s
E2	17.37 s	1.84 s	19.21 s
E3	4.39 s	5.14 s	9.53 s
Logar	813.79 s	52.47 s	866.26 s
Oaku	2.39 s	1.60 s	3.99 s
M	0.59 s	0.37 s	0.96 s

We remark that the parenthesized values in the Gerdt and Butcher cases give the timing after a partial hand-factoring of the input. The utility of a factoring Buchberger algorithm is self-evident.

We have compared our algorithm with the macaulay2 implementation of the direct algorithm. For most examples we have not been able to compute the radical with the direct algorithm, while for the others the timings of our algorithm are better, see [CCT]. More experiment is needed, since further optimization of both algorithms may shift the balance. After the development of a C implementation of our algorithm we plan to compare the timings with the direct algorithm as then implemented in macaulay2.

Remark that in this binomial ideal all the computations have been with binomials; any other algorithm with any form of generic (or random) projection direction would have destroyed the format. At least for binomial ideals the advantage over algorithms using changes of coordinates is obvious. See [ES] for more information on binomial ideals.

5.6 An Example

We show how our algorithm behaves to find the radical of the example of D. Jaffe, reported in [EHV], in which their algorithm fails in characteristics 7. The example, that can be generalized in any characteristics, is $(z^7 - xyt^5, y^4 - x^3t)$.

The example is of dimension 2, hence it is a complete intersection, and pure dimensional, but we want to forget it and compute the example completely. We will use block orderings, and DegRevLex in every block.

The set of generators is already Gröbner with respect to several orderings (DegRevLex, Lex with $z > y > x > t$, etc.).

We have to choose a variable to eliminate; if we choose z, then the ideal is already Gröbner and $g = y^4 - x^3t$, hence no progress is made. If nilpotents exist, they are along the z direction.

Then we change elimination variable; if we choose x, (but with other variables it is not much different) then a Gröbner basis consists of

$$(x^3t - y^4, xyt^5 - z^7, x^2z^7 - y^5t^4, xz^{14} - y^6t^9, z^{21} - y^7t^{14})$$

hence $g = z^{21} - y^7t^{14} = (z^3 - yt^2)^7$, and its radical is $z^3 - yt^2$.

Compute now the Gröbner basis of $(z^7 - xyt^5, y^4 - x^3t, z^3 - yt^2)$. It is composed of 17 elements, but we just need two: $g = z^3 - yt^2$ and $h_1 = xyt^5 - y^2zt^4$ (we use the notation of section 5.3.1). We have $g_1 = yt^5$.

Now we saturate $(z^7 - xyt^5, y^4 - x^3t, z^3 - yt^2)$ with respect to yt^5, and we obtain $(z^3 - yt^2, xt - yz, x^2z - y^3, xz^2 - y^2t)$. We know now that this is the top-dimensional reduced component. If we saturate $(z^7 - xyu^5, y^4 - x^3u)$ with it, we obtain 1, (this can be checked, instead of saturating, showing that the seventh power of every generator is inside the input ideal) hence no other component exists. The radical is hence $(z^3 - yt^2, xt - yz, x^2z - y^3, xz^2 - y^2t)$.

Further developments

The results reached in this thesis has been resumed in the introduction. Here we describe briefly some open question and the prospects for the further development of this work.

Optimization of the Buchberger algorithm

The extension of the Hilbert-Driven techniques to the multigraded case leads to substantial improvements of the computational costs. The only drawback is the meager supply of naturally multigraded problems. A systematic examination on the possibility of multigraduations introduction in the basic operations has to be done. Moreover, since the description of the multigraded Poincaré series for high-type system of weights may be surprisingly complex, an ad hoc algorithm for the computation of only the multidegree(s) we are interested in should be developed. The algorithms now in use seems to offer only marginal prospects for improvements.
The Dynamic Buchberger algorithm developed in the thesis is quite efficient on the binomial examples (but here ad hoc techniques [BCNR] are substantially better than the straightforward Gröbner basis computation), and it seems to have good prospects in three cases

problems with an high number of polynomials whose support is small;

large, yet unsolved problems, where a careful (automatic) choice of the ordering may give a chance of success.

A yet unsolved problem is the instability of the heuristics based on Hilbert Function considerations. Further testing is necessary after a new, state-of-the-art implementation.

Module implementation through embedding

The relative efficiency of the embedding approach with respect to the other approaches has still to be assessed. This comparison should be done after the full development of the polynomial arithmetic techniques of Capani PhD. thesis [C].
An open opportunity is the development of a dynamic version of the resolution algorithm which may minimize the size of the Gröbner bases of syzygy modules.

Optimization of ideal/module operations

For the transporter and saturation cases, we have developed new algorithms derived from the tag variable module intersection algorithm (Hilbert-Driven) that seem to give the best performances, but an exhaustive comparison of the different algorithms has still to be done, since the interaction of their variants gives a quite high number of different algorithms (more than hundred for the saturation). If further information (e. g. we know $I + J$ or at least its Hilbert function) are available, the double Hilbert driven algorithm sketched in chapter 4 promises to be quite efficient. Further experiments are needed.

Ideal decomposition

The ideal decomposition algorithm discussed in this thesis has to be carefully compared to the algorithm in [EHV] and [LS]. Moreover, the utility of an a priori equidimensional decomposition has to be assessed for primary decompositions. Some comparison with the algorithms implemented in `Singular` [Singular] support this idea. It would be interesting to develop an integration of the two algorithms.

Bibliography

[A] T.M. Apostol, *Introduction to Analytic Number Theory*, Springer Verlag, 1976.

[AGK] B. Amrhein, O. Gloor, W. Küchlin, *On the Walk*, Theoretical Computer science 187 pp.179-202, 1997.

[AM] M.F. Atiyah, I.G. Macdonald, *Introduction to Commutative Algebra*, Addison-Wesley, 1969.

[AMR] M. Alonso, T. Mora, M. Raimondo, *Local decomposition algorithms*, AAECC90, Tokyo, LNCS 508, pp.208-211, 1990.

[AT] D. Attardi, and C. Traverso, *A strategy accurate parallel Buchberger Algorithm*, Journal of Symbolic Computation, 21, 4-6, pp.411-425, 1996.

[BGS] D. Bayer, A. Galligo, M. Stillman, *Gröbner bases and extensions of scalars*, in: D. Eisenbud, L. Robbiano, *Computational methods in Algebraic Geometry and Commutative Algebra*, Symposia Mathematica 24, 1993.

[BS] D. Bayer and M. Stillman, *Computation of Hilbert functions*, Journal of Symbolic Computation **14**, pp 31-50, 1992.

[BS2] D. Bayer and M. Stillman, *A theorem on refining division orders by the reverse lexicographic oredering*, Duke J. Math. 55, pp.321-328, 1987.

[BM] D. Bayer and D. Mumford, *What can be computed in Algebraic Geometry?*, Cambridge University Press, Symposia Math. XXXIV pp.1-48, 1992.

[BW] T. Becker and V. Weispfenning, *A computational approach to Commutative Algebra*, Graduate Text in Mathematics, Springer Verlag, 1992.

[B] A. M. Bigatti,*Computation of Hilbert-Poincaré Series*, Journal of Pure and Applied Algebra, **119**, 3, pp.237-253, 1997.

[BCR] A.M. Bigatti, M. Caboara and L. Robbiano *On the Computation of Hilbert-Poincarè Series*, AAECC 2, pp.21-33, 1991.

[BCRT] A.M. Bigatti, P. Conti, L. Robbiano, and C. Traverso, *A "Divide and conquer" algorithm for Hilbert-Poincaré series, multiplicity and dimension of monomial ideals*, Proceedings of AAECC-10 LNCS 673, pp.76-88, Springer, 1992.

[BCNR] A.M. Bigatti, A. Capani, G. Niesi and L. Robbiano, *Hilbert Poincarè Series and Eliminations problems*, submitted, 1998.

[BCNR] A.M. Bigatti, R. La Scala and L. Robbiano, *Computing Toric Ideals*, submitted, 1998.

[BGK] W. Boege, R. Gebauer, H. Kredel, *Some examples for solving systems of algebraic equations by calculating Gröbner bases*, Journal of Symbolic Computation **2**, pp.83-89, 1987.

[Bu] B. Buchberger, *Gröbner bases: An algorithmic method in polynomial ideal theory*, In: Bose, N.K.(ed) *Multidimensional Systems Theory*, Riedel, Dordrecht, pp.184-232, 1985.

[Ca] M. Caboara, *A Dynamic Algorithm for Gröbner basis computation*, ISSAC93, Kiev, pp.275-283, 1993.

[Ca1] M. Caboara, *A modified Buchberger Algorithm for resolutions*, Preprint (1997)

[CDR] M. Caboara, G. De Dominicis and L. Robbiano, *Multigraded Hilbert functions and Buchberger algorithm*, Proceeding of the ISSAC96 Conference, Zurich, pp.72-85, 1996.

[CS] M. Caboara and M. Silvestri, *Classification of compatible module orderings*, Submitted (1997).

[CS1] M. Caboara and M. Silvestri *Compatible module orderings*, ISSAC'96 Poster Session, pp.17-22, 1996.

[CT1] M. Caboara and C. Traverso, *How to compute an intersection*, Preprint (1997).

[CCT] M. Caboara, P. Conti, and C. Traverso, *Yet Another Ideal Decomposition Algorithm*, AAECC-12, Springer LNCS 1255, pp.39-54, 1997.

[C] A. Capani, Phd Thesis, Dipartimento di Informatica, Università di Genova, in prep. (1997).

[COCOA] A. Capani, G. Niesi and L. Robbiano *CoCoA, a system for doing Computations in Commutative Algebra*, Dipartimento di Matematica, Università di Genova, Available via anonymous ftp from `cocoa.dima.unige.it`. Web page `http://cocoa.dima.unige.it`.

[CN] A. Capani and G. Niesi,framework for a family of Buchberger-like algorithms, in *Gröbner bases and Applications* (Proceedings of the 33 Years of Gröbner bases), Buchberger and Winkler eds., Cambridge University Press, vol 251, 338-350.

[CDNR] A. Capani, G. de Dominicis, G. Niesi and L. Robbiano, *Computing Minimal Resolutions*, Journal of Pure and Applied Algebra, Vol. 117-118, pp.105-117, 1997.

[CSi] G. Carra' Ferro and W.Y. Sit, *On term orderings and rankings*, in Colloquia, Computational Albegra, Lecture Notes Pure and Applied Math. 151, pp.31-77, Dekker, NewYork, 1994.

[CT] P. Conti, C. Traverso, *A Case of Automatic Theorem Proving in Euclidean Geometry: the Maclane* 8_3 *Theorem*, Proc. AAECC-11 LNCS 948, pp.183-193, 1995.

[CLS] D. Cox, J. Little and D. O'Shea, *Ideal, Varieties and Algorithms*, Undergraduate Text in Mathematics, *Springer Verlag*, 1995.

[D] J.H. Davenport, *Polynomial factorization*, MEGA-96, J. Pure Appl. Algebra, pp.117-118, 1996.

[E] D. Eisenbud, *Commutative Algebra with a View Toward Algebraic Geometry*, Springer Verlag, Graduate Text in Mathematics, 1994.

[EHV] D. Eisenbud, C. Huneke and W. Vasconcelos, *Direct methods for primary decomposition*, Inventiones Math, 110, pp.207-235, 1992.

[ES] D. Eisenbud and B. Sturmfels, *Binomial ideals*, Duke Math. J. 84, pp.1-45 1996.

[FGLM] J.C. Faugère, P. Gianni, D. Lazard, T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, Journal of Symbolic Computation, **16**(4), pp.329-344, 1993.

[G] P. Gianni, *Properties of Gröbner bases under specialization*, Eurocal 1987, LNCS 378, pp.293-297, 1988.

[GTZ] P. Gianni, B. Trager and G. Zacharias, *Gröbner bases and primary decomposition of polynomial ideals*, Journal of Symbolic Computation, **6**/2/3 pp.149-167, 1988.

[GMNRT] A. Giovini, T. Mora, G. Niesi, L. Robbiano and C. Traverso, *"One sugar cube, please", or Selection strategies in the Buchberger Algorithm*, ISSAC91, ACN, 1991.

[GM] R. Gebauer and M. Möller, (On an installation of the Buchberger algorithm, Journal of Symbolic Computation, **6**/2/3 pp.275-286, 1988.

[GGMNPPSS] H. Grassman, G.M. Greuel, B. Martin, W. Neumann, G.Pfister, W. Pohl, H. Schonemann and T. Siebert, *Standard bases, syzygies and their implementation in SINGULAR*, Preprint, (1997).

[GS] Gritzmann, P. and B. Sturmfels (1990), *Minkowski addition of polytopes: computational complexity and applications to Grobner bases*, SIAM-J.-Discrete-Math, 6, no. 2, pp.246-269, 1993.

[K1] M. Kalkbrener, *Solving systems of algebraic equations by using Gröbner bases*, Eurocal 1987, LNCS 378, pp.282-292, 1988.

[K2] M. Kalkbrener, *A generalized euclidean algorithm for computing triangular representations of algebraic varieties*, Journal of Symbolic Computation, 15, pp.143-167, 1993.

[K3] M. Kalkbrener, *Prime decomposition of radicals in polynomial rings*, Journal of Symbolic Computation, 18, pp.365-372, 1994.

[Ls] R. La Scala, *An algorithm for complexes*, Proceedings of ISSAC'94, pp.264-268, 1994.

[LS] R. La Scala and M. Stillman, *Strategies for computing minimal free resolutions*, Preprint (1997).

[L] D. Lazard, *A new method for solving algebraic systems of positive dimension,* Discr. App.Math, 33, pp.147-160,1991.

[M] Matsumura (1986), *Commutative Algebra*, Cambridge University Press, Cambridge, 1970.

[Metc] F. Mora la dimostrazione della complessita' della GB + alg. omogeneo.

[MM] H.M. Möller and F. Mora, *New constructive methods in classical ideal theory*, J. Algebra **100**, pp.138-178, 1986.

[MMT] T. Mora, M. Möller and C. Traverso, *Gröbner basis computation using syzygies*, ISSAC '91, 1991.

[Mo] T. Mora, *An introduction to commutative and noncommutative Gröbner bases*, Theoretical Computer Science, 134, no.1, pp.131-173, 1992.

[MR] F. Mora and L. Robbiano, *The Gröbner fan of an Ideal*, Journal of Symbolic Computation, 6, no.2-3, pp.183-208, 1988.

[Mu] D. Mumford, *Lectures on Curves on an Algebraic Surface*, Annals of Math. Studies, 1966.

[PoSSo] Esprit project n. 6846 Posso.

[Ri] C. Riquier, Les systèmes d'équation aux d́erivées partielles, Gauthier-Villars, Paris, 1910.

[R1] L. Robbiano, *Term orderings on the polynomial rings*, EUROCAL '85, SpringerLNCS **204**, pp.513-517, 1985.

[R2] L. Robbiano, *Dispense*, (1997).

[R3] L. Robbiano, *Introduction to the theory of Hilbert functions*, In Anthony V. Geramita, Ed., *The Curves Seminar at Queen's*, vol VIII, pp.B1-B26, Kingston, Ontario, Queen's Papers in Pure and Applied Mathematics, No. 85, 1990.

[Ro] Rothenberg, *Linear Programming*, New York, Elsevier-North Holland, 1979.

[RR] C.J. Rust and G.J. Reid, *Rankings of partials derivatives*, Proceedings of ISSAC'97, pp.9-16, 1997.

[Ru] C.J. Rust, *The classification of Rankings of partials derivatives*, PhD thesis, in prep. (1997).

[RV] R. Robbiano and G. Valla, *Hilbert Poincarè Series of Bigraded Algebras*, Preprint, 1998.

[Ru2] C.J. Rust, *On the classification of Rankings of partials derivatives*, May be downloaded from `http://www.math.uchicago.edu/~rust/orderings.ps`.

[S] R.P. Stanley, *Combinatorics and Commutative Algebra*, Second Edition, Birkhäuser, PM 41, 1996.

[Singular] G.M. Greuel, G. Pfister and H. Schonemann, *Singular, A Computer Algebra System for Singularity Theory, Algebraic Geometry and Commutative Algebra.* The Singular system and its user manual and tutorials are available by anonimous ftp from the web page `ftp://www.mathematik.uni-kl.de/pub/Math/Singular`.

[Sh] F.O. Schreyer, *Die Berchnung von Syzygien mit dem verallgemeinerten Weierstrass'schen Divisionsatz*, Diplomarbeit, Hamburg, 1980.

[Si] Th. Siebert, *On strategies and implementations for computations of free resolutions*, Preprint Nr. 8 des Zentrums fuer Computeralgebra, Universitaet Kaiserslautern, 1996.

[Sv] M. Silvestri, *Thesis*, Dipartimento di matematica, Università degli studi di Pisa, (1993).

[Sp] D.A. Spear, *A constructive approach to commutative ring theory*, In: *Proc, MACSYMA User's conference*, NASA CP-2012, pp.369-376, 1989.

[St] B. Sturmfels, *Dynamic version of the Buchberger Algorithm*, Preprint (1989).

[Sw] F. Schwartz, *Monomial orderings and Gröbner Bases*, ACM SIGSAM Bullettin 25, No 1, pp.10-23, 1991.

[SY] T. Shimoyama and K. Yokoyama, *Localization and primary decomposition of polynomial ideals*, Journal of Symbolic Computation, 22, pp.247-277, 1996.

[T] C. Traverso, *Hilbert Functions and the Buchberger Algorithm*, Journal of Symbolic Computation, 22-4 pp.355-376, 1996.

[TD] C. Traverso and L. Donati, *Experimenting the Gröbner basis algorithm with the ALPI system*, ISSAC89, ACM press, NewYork, pp.192-198, 1989.

[Th] J.M. Thomas, *Matrices of Integers Ordering Derivatives* Trans. Amer. Math Soc. **33**, pp.389-410, 1931.

[V] W.V. Vasconcelos, *Jacobian matrices and constructions in Algebra*, AAECC-9, LNCS 539, pp.48-64, 1991.

[Wa] B. Wall, *On the computation of syzygies*, ACM SIGSAM Bullettin **23**/4 pp.5-14, 1989.

[W] V. Weispfenning, *Admissible orders and linear forms*, *ACM SIGSAM Bullettin* **21**/*2, pp.16-18*, 1987.

[W1] V. Weispfenning, *Differential Term Orderings*, ACM SIGSAM Bullettin **21**/2, pp.16-18, 1992.

[ZS] O. Zariski, and P. Samuel *Commutative Algebra*, vol I-II, New York, Van Nostrand, 1958-1960.