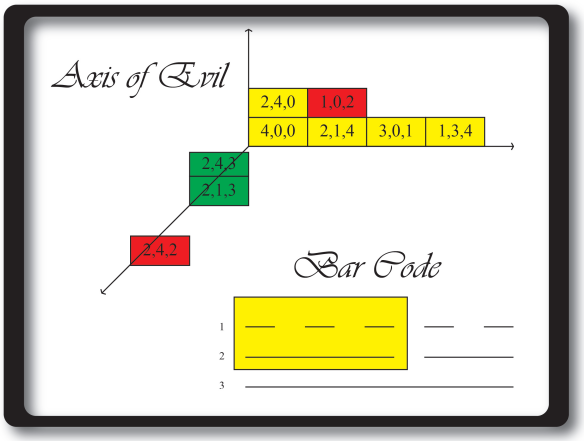
Combinatorial structure of monomial ideals. Michela Ceria

Michela Ceria

Università degli Studi di Torino
Dipartimento di Matematica

Scuola di Dottorato in Scienza ed Alta Tecnologia
Ciclo XXVI



Combinatorial structure of monomial ideals.

Michela Ceria

Tutors: Prof. Maria Grazia Marinari
Prof. Margherita Roggero

Coordinatore del Dottorato: Prof. Ezio Venturino
Anni Accademici: 2011–2013
Settore Scientifico-disciplinare di afferenza: Matematica

Università degli Studi di Torino
Scuola di Dottorato in Scienza ed Alta Tecnologia

Tesi di Dottorato di Ricerca in Scienza ed Alta Tecnologia
Indirizzo: Matematica

COMBINATORIAL STRUCTURE OF MONOMIAL IDEALS.

Michela Ceria

Tutors: Prof. Maria Grazia Marinari
Prof. Margherita Roggero

XXVI ciclo – Gennaio 2014

W.r.t. Lex, $a < b < ... < z$.

To Margherita, Maria Grazia and Teo.

Abstract

Combinatorial structure of monomial ideals.

Michela Ceria – XXVI ciclo

Scuola di Dottorato in Scienza ed Alta Tecnologia
Università degli Studi di Torino – Gennaio 2014

In this Thesis, we study monomial ideals from a combinatorial point of view.
We are mainly interested in the structure of the associated Groebner escalier but, sometimes, we have also to deal with the initial ideal.
First of all, we examine all the existing combinatorial methods to compute the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$ associated to the zerodimensional radical ideal $I(\mathbf{X})$ of a finite set of distinct points $\mathbf{X}$. More precisely, we start from Cerlienco-Mureddu correspondence and we examine the other methods which came up later on, such as Gao-Rodrigues-Stroomer method, Lederer's variation and Lex Game.
Next, we face the problem of constructing a linear factorization of a minimal Groebner basis for a zerodimensional radical ideal. The existence of such a factorization has been stated and proved by Maria Grazia Marinari and Teo Mora, in the *Axis of Evil Theorem* [2, 69, 70]. In

this Thesis we give an alternative constructive proof, together with an algorithm computing concretely the factorization and we study deeply the structure of the Groebner escalier, in connection to the Axis of Evil factorization.
Then, we develop a visual language in order to represent finite sets of terms and infinite order ideals via bidimensional pictures, the Bar Codes.
We show that the pictures we get allow us to read easily many properties of the monomial ideal (expecially connected to Janet decomposition for terms [54, 55, 56, 57]) and to develop an iterative version of the Axis of Evil algorithm.
Thanks to the Bar Code structure, moreover, we are able to connect commutative algebra and enumerative combinatorics, by giving a bound for the number of strongly stable ideals with a fixed constant affine Hilbert polynomial, by putting them in biunivocal correspondence with plane partitions.
Finally, we show how the Axis of Evil theorem can be applied to coding theory, more precisely to the decoding procedure for binary BCH codes and to the computation of sparse general error locator polynomials.

Contents

List of Figures

List of Tables

Introduction

This thesis is centered on an exam of the combinatorial structure of both the *initial ideal* and the *Groebner escalier* of an ideal of the ring of polynomials.
Many properties of an ideal I can be deduced by studying its initial ideal with respect to some term ordering. The initial ideal is a monomial ideal, namely it has a generating set only composed of terms and it is possible to recover the monomial basis of the quotient algebra (the Groebner escalier) from it.
In the case of a zerodimensional radical ideal I, namely the ideal of a finite set of distinct points $\mathbf{X}$, the Groebner escalier $\mathsf{N}(I)$ is a finite set.
Clearly it is possible to recover it from the initial ideal, but this is rather ineffective. Indeed, in order to get the initial ideal it is necessary to compute the Groebner basis of I from some generating set of polynomials. The computation is performed via Buchberger algorithm, and it is well known that this algorithm is heavy from a computational point of view. The first mathematicians who dealt with this problem are Buchberger and Moeller in [12] (1982). In the cited paper, they developed a *polynomial* algorithm which computes the reduced lexicographical Groebner basis of a zerodimensional radical ideal via *interpolation* on the finite set of distinct points representing the associated variety. Apart from the Groebner basis, the algorithm provides also the terms in the lexicographical Groebner escalier.
A few years later, Cerlienco and Mureddu [20, 21, 22] developed a combinatorial algorithm,

computing the lexicographical Groebner escalier *directly* from the points of $\mathbf{X}$, exploiting a series comparisons among the coordinates. This algorithm provides a *biunivocal correspondence* between the points in $\mathbf{X}$ and the terms in the Groebner escalier: the so called Cerlienco-Mureddu Correspondence.
Next, other methods, optimizing Cerlienco-Mureddu algorithm, have been developed, for instance by Felszeghy-B. Ráth-Rónyai, Gao-Rodrigues-Stroomer and Lederer.
We give an overall view of these methods, equipped with detailed examples.
Thanks to the structure of the (finite) lexicographical Groebner escalier, it is possible to examine also the structure of the zerodimensional radical ideal defining a given finite set of distinct points $\mathbf{X}$.
Via the so called *Axis of Evil theorem*, M.G. Marinari and T. Mora enhanced the classical *Lazard structural theorem* to the case of $n > 2$ variables. The Axis of Evil theorem assures, for a minimal lexicographical Groebner basis of a zerodimensional radical ideal, the existence of a factorization *linear* in the leading terms.
The Axis of Evil theorem is one of the main topics of this thesis.
We will give a computational proof of the theorem, providing an interpolation algorithm *à la Moeller*, which computes the above factorization (called *Axis of Evil factorization*) and then we will give some variations of the aforesaid algorithm. Moreover, we give another combinatorial method to compute the Groebner escalier, providing an ordering on the terms and on the corresponding points which makes the interpolation simpler.
The Axis of Evil factorization can be applied to the field of *coding theory*. More precisely, we deal with the decoding of *BCH codes*, in the realm of the so called *Cooper's philosophy* [28, 29], which introduces the use of Groebner bases for decoding.
Starting from the works by Chen [23, 24, 25], Cooper's ideas have been improved by introducing and studying the *syndrome variety* in order to optimize the decoding process. In this context are also placed many interesting works by Mora, Orsini and Sala [78, 82, 83], from which arises the application of the Axis of Evil theorem to decoding BCH codes. In these papers, the *general error locator polynomial*, whose roots are the exactly the error locations, is introduced.
Sparsity of this polynomial would be rather important for pratical applications and it would be appreciable if such polynomial grew linearly with the cardinality if the base field $\mathbb{F}_q$ over which the code is defined.
In a joint work with M. Sala and T. Mora, we exploit the Axis of Evil factorization to find a sparse general error locator polynomial, minimizing the number of points to work with and computing the structure of the associated Groebner escalier.
We will see that encouraging results can be found in some simplified case. The points con-

figurations we get, turn out to have a very precise structure, connected to the cycles of the base field.
Since it is a work in progress, we will give only partial results for the mentioned cases.

Studying the Groebner escalier, the necessity to represent it visually arose. There are some graphical representation of the Groebner escalier in literature, but they are rather complicated to draw if the cardinality of $\mathsf{N}(I)$ is a big number (and impossible in the case of an infinite $\mathsf{N}(I)$) or if the number of involved variables is higher than five.
In this thesis we develop a simple bidimensional representation for finite and infinite Groebner escaliers, called *Bar Code diagram*. Such a diagram is also simple to encode in a computer, so it can be useful from a computational point of view.
First of all, it enabled us to find a new combinatorial method for the Groebner escalier, analogous to the aforesaid ones and enjoying many of their best features. Secondly, it gave us the possibility to find an iterative algorithm to compute the Axis of Evil factorization of a minimal lexicographical Groebner basis for the ideal of a finite set of distinct points.

Moreover, studying the shape of the Bar Code diagram for *strongly stable* monomial ideals with constant affine Hilbert polynomial we noticed that the diagrams are joined by a sort of "pattern".
Examining it, we started connecting objects belonging to different fields, namely: strongly stable ideals (from commutative algebra), and plane partitions of integer numbers (from enumerative combinatorics). This work is still in progress and we display here only partial results, namely the ones for strongly stable ideals in two or three variables, with constant affine Hilbert polynomial.
For the case of two variables, we have proved a biunivocal correspondence between strongly stable ideals and integer partitions of p, so we are able to count exactly their number. For three variables, instead, we have proved the biunivocal correspondence between strongly stable ideals and some particular plane partitions, for whose number, for now, we only have an upper bound.

Finally, exploiting the properties of the generating sets of monomial ideals, it is possible to deal with the following

Problem 0.0.1. *Given any monomial ideal $J \lhd \mathcal{P} := \mathbf{k}[x_1, ..., x_n]$, find a characterization for the family $\mathcal{M}f(J)$ of all homogeneous ideals $I \lhd \mathcal{P}$ such that the basis of $\mathcal{P}/I$ is given by the set of terms in the Groebner escalier $\mathsf{N}(J)$ of J.*

This problem has been deeply analyzed in [8, 27, 64] for the case J strongly stable ideal, which is also the most suitable case for studying the Hilbert scheme. In [8, 27], the families of the form $\mathcal{M}f(J)$ for J strongly stable have studied, giving also computational methods to deal with them.
In a joint work with T. Mora and M. Roggero [19], we generalize the problem above to arbitrary monomial ideals on the polynomial ring with coefficients in a commutative ring.
In order to give such a generalization, we exploit and enhance some concepts introduced by Janet [54, 55, 56, 57], such as the definition of *multiplicative variable* and the one of *complete system*, leading to the so called *Janet decomposition* for terms.
Starting from the generating set of a monomial ideal, Janet gives a very precise decomposition of the ideal itself (and also of its Groebner escalier). In Janet's theory the ideals are generated by the so called *involutive bases*.
If we draw the Bar Code of a finite set of terms (not necesarily an order ideal) we can answer some combinatorial problems on Janet decomposition. For example, we can detect the multiplicative variables or decide on the completeness of a system.
We have to point out that Janet gave two different definitions of multiplicative variable, presented in [54, 55] and in [56], totally equivalent if we are in general coordinates. In [19], we compare them and we introduce the notion of *stably complete* set of terms, indicating sets for which both conditions hold. Each monomial ideal J has one and only one stably complete set of generators (possibly made of infinitely many terms) that we call *star set* and denote by $\mathcal{F}(J)$. The star set can be computed from the Groebner escalier of J using again the Bar Code structure in a very simple way. Furthermore, in analogy with [8, 27] we define a reduction procedure with respect to a homogeneous set of polynomials marked on a stably complete system proving its noetherianity.
The most interesting cases are the ones involving ideals with finite stably complete generating set , i.e. the *quasi stable* ideals, whose star set is exactly the Pommaret basis. Note that a monomial ideal is *stable* if and only if its star set coincides with the monomial basis.
Properties of the star set allowed us to provide a new version of Moeller algorithm which computes a lexicographical involutive basis for the zerodimensional radical ideal of a finite set $\mathbf{X}$ of distinct points via interpolation on the elements of $\mathbf{X}$.
During my PhD I worked under the supervision of Professors M.G. Marinari, T. Mora and M. Roggero, cooperating also with Professors F. Cioffi, W. Decker and H. Schoenemann. With these last ones I implemented, using the the computer algebra system Singular [30], two libraries which are part of version 3-1-6 of the software. In this thesis I explain how the implementaton has been made and we also attach the source code.

In chapter 1, we give all the notation needed in the whole thesis, involving polynomials and Groebner bases, Groebner duality and Macaulay bases, Graph Theory, expecially trees and tries, and we recall the main features of Moeller algorithm. We also define the existing visual representations for terms: the *tower structure* and the diagrams introduced by M.G. Marinari and L. Ramella.
Chapter 2 is devoted to the study of all the combinatorial methods for computing the Groebner escalier of the ideal of a finite set of distinct points. We start with Cerlienco-Mureddu correspondence, then we examine Gao-Rodrigues-Stroomer method with the variation proposed by Lederer and finally the Lex Game algorithm by Felszeghy-B. Ráth-Rónyai.
In chapter 3, after explaining Lazard's algorithm for monomial bases, the Macaulay's trick and Lazard structural theorem, we introduce the Axis of Evil theorem by Marinari and Mora and the associated algorithm. This algorithm gives a simple proof for Marinari-Mora theorem.
The whole chapter 4 describes a new version of the Axis of Evil algorithm, under suitable hypotheses. In order to make the interpolation process simpler, we define an interpolation oriented alternative to the algorithms described in chapter 2.
In chapter 5 we first define the Bar Code of a finite set of terms, studying its main features. Then we define the star set in terms of Bar Codes (so from the Groebner escalier point of view), proving its characterization in terms of generating sets. After that, we extend both the notion of Bar Code and of star set to infinte Groebner escaliers. We give then some applications, such as another alternative algorithm to the ones of chapter 2 and a related iterative version of the Axis of Evil algorithm. Moreover, we present some first results in enumerative combinatorics for strongly stable ideals.
In chapter 6, we first recall the theory developed in [8, 27] for J-marked families, explaining how it leads to the Singular libraries we implemented. After that, we deal with Janet decomposition for terms, relating the problem to the Bar Code structure of the generating set for a monomial ideal.
After defining the star set, we characterize stable and quasi stable ideals and we define the notherian reduction procedure for homogeneous polynomials, marked on a stably complete set. Moreover, we study J-marked families using the reduction procedure.
We give then an historical note on the concepts by Janet we exploited and, at the very end, we describe the Moeller version which computes an involutive basis in the zerodimensional radical case.
Dulcis in fundo, chapters 7 and 8 are devoted to apply the Axis of Evil algorithm to coding theory.
More precisely, chapter 7 starts giving the most important notions of error correcting codes

and then it focuses on cyclic and BCH codes, by treating Cooper's decoding philosophy, Chen's works on the *syndrome variety* and all the improvements by T. Mora, E. Orsini and M. Sala, introducing the concept of *general error locator polynomial*.
On the other hand, chapter 8, treats the decoding process for BCH codes by determining the general error locator polynomial and showing how the structure of the Groebner escalier and the Axis of Evil algorithm can help in finding a sparser locator.
Finally appendix A and B contain respectively the Singular code of our libraries and the data obtained by computing the locator polynomials and the related points configurations.

Part I

Getting started.

CHAPTER 1

Notation and preliminaries.

1.1 Polynomials and Groebner bases.

In this thesis, we follow the notation of [77, 79].
We let $\mathcal{P} := \mathbf{k}[x_1, ..., x_n]$ the graded ring of polynomials in n variables with coefficients in the field $\mathbf{k}$.
We usually denote by $\mathcal{S} := \mathbf{k}[x_0, ..., x_n]$ the ring of polynomials in $n+1$ variables and coefficients in the base field $\mathbf{k}$.
The *semigroup of terms*, generated by the set $\{x_1, ..., x_n\}$ is:

$$\mathcal{T} := \{x_1^{\alpha_1} \cdots x_n^{\alpha_n},\ (\alpha_1, ..., \alpha_n) \in \mathbb{N}^n\}.$$

Denoting $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, we define $deg(\tau) = \sum_{i=1}^{n} \alpha_i$, the degree of τ and, for each $h \in \{1, ..., n\}$ $deg_h(\tau) := \alpha_h$ is the h-degree of τ.

For each $d \in \mathbb{N}$, $\mathcal{T}_d$ denotes the d-degree part of $\mathcal{T}$, and for each $M \subseteq \mathcal{T}$, $M_d = M \cap \mathcal{T}_d$, whereas $\mathcal{T}(d)$ is the degree $\leq d$ part of $\mathcal{T}$, with $|\mathcal{T}_d| = \binom{n+d-1}{d}$. We use analogous notation for $\mathcal{P}$, observing that by abuse of notation we also denote by $\mathcal{P}(d)$ the vector space gener-

ated by $\mathcal{T}(d)$.
Letting $\alpha = (\alpha_1, ..., \alpha_n) \in \mathbb{N}^n$, we will often write x^α instead of $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$.
We define also the set

$$\mathcal{T}[m] := \mathcal{T} \cap \mathbf{k}[x_1, ..., x_m] = \{x_1^{a_1} \cdots x_m^{a_m} / \, (a_1, ..., a_m) \in \mathbb{N}^m\}.$$

A *semigroup ordering* $<$ on $\mathcal{T}$ is a total ordering such that $\tau_1 < \tau_2 \Rightarrow \tau\tau_1 < \tau\tau_2$, $\forall \tau, \tau_1, \tau_2 \in \mathcal{T}$.
A semigroup ordering is called *inf-limited* if:

- $x_i < 1$, for each $i \in \{1, ..., n\}$;
- for each infinite decreasing sequence in $\mathcal{T}$, $\tau_1 > ... > \tau_j > ...$ and each $\tau \in \mathcal{T}$ there is an $r \in \mathbb{N}$, with $\tau_r < \tau$.

For each semigroup ordering $<$ on $\mathcal{T}$, we can represent a polynomial $f \in \mathcal{P}$ as a linear combination of terms arranged w.r.t. $<$, with coefficients in the base field $\mathbf{k}$:

$$f = \sum_{\tau \in \mathcal{T}} c(f, \tau)\tau = \sum_{i=1}^{s} c(f, \tau_i)\tau_i : \, c(f, \tau_i) \in \mathbf{k}^*, \, \tau_i \in \mathcal{T}, \, \tau_1 > ... > \tau_s,$$

with $\mathsf{T}(f) = Lt(f) := \tau_1$ the *leading term* of f, $Lc(f) := c(f, \tau_1)$ the *leading coefficient* of f, $Lm(f) = \mathsf{M}(f) := c(f, \tau_1)\tau_1$ the *leading monomial* of f and $tail(f) := f - c(f, \mathsf{T}(f))\mathsf{T}(f)$ the *tail* of f.
Letting $\delta := deg_n(f)$ the degree of f w.r.t. x_n we can write uniquely

$$f = \sum_{i=0}^{\delta} g_i x_n^i \in \mathbf{k}[x_1, ..., x_{n-1}][x_n], \, g_i \in \mathbf{k}[x_1, ..., x_{n-1}], \, g_\delta \neq 0$$

denoting by $Lp(f) := g_\delta$ the *leading polynomial* of f and by $\mathsf{T}p(f) = g_0$ the *trailing polynomial* of f w.r.t n.

For each term $\tau \in \mathcal{T}$ and $x_j | \tau$, the only $\upsilon \in \mathcal{T}$ such that $\tau = x_j \upsilon$ is called j-th *predecessor* of τ.

A subset $\mathsf{N} \subseteq \mathcal{T}$ is an *order ideal* if $\tau \in \mathsf{N} \Rightarrow \sigma \in \mathsf{N} \, \forall \sigma | \tau$[1]. A subset $\mathsf{N} \subseteq \mathcal{T}$ is an order ideal if and only if $\mathcal{T} \setminus \mathsf{N} = J$ is a semigroup ideal (i.e. $\tau \in J \Rightarrow \sigma\tau \in J$, $\forall \sigma \in \mathcal{T}$).
For each semigroup ideal $J \subset \mathcal{T}$, $\mathsf{N}(J) := \mathcal{T} \setminus \mathsf{T}(J)$ and its monomial basis $\mathsf{G}(J)$ satisfies the conditions below

$$\mathsf{G}(J) := \{\tau \in J \mid \text{ each predecessor of } \tau \in \mathsf{N}(J)\} =$$
$$= \{\tau \in \mathcal{T} \mid \mathsf{N}(J) \cup \{\tau\} \text{ order ideal}, \, \tau \notin \mathsf{N}(J)\}.$$

[1]The corresponding notion for $\mathbb{N}^n$ is named *Ferrers diagram*.

For all subsets $G \subset \mathcal{P}$, $\mathsf{T}\{G\} := \{\mathsf{T}(g), g \in G\}$ and $\mathsf{T}(G)$ is the semigroup ideal $\{\tau\mathsf{T}(g), \tau \in \mathcal{T}, g \in G\}$.

We define also $\mathsf{M}\{G\} := \{\mathsf{M}(g), g \in G\}$ and $\mathsf{M}(G) := \{\mathsf{M}(a\tau g), a \in \mathbf{k}^*, \tau \in \mathcal{T}, g \in G\}$. For any ideal $I \lhd \mathcal{P}$ the monomial basis of the semigroup ideal $\mathsf{T}(I) = \mathsf{T}\{I\}$ is called *monomial basis* of I, the ideal $In(I) := (\mathsf{T}(I))$ is the *initial ideal* and the *border set* of I is:

$$\mathsf{B}(I) := \{x_h\tau, 1 \leq h \leq n, \tau \in \mathsf{N}(I)\} \setminus \mathsf{N}(I) =$$
$$= \mathsf{T}(I) \cap (\{1\} \cup \{x_h\tau, 1 \leq h \leq n, \tau \in \mathsf{N}(I)\}).$$

If $I := (G)$ we have $\mathsf{M}(I) := \mathsf{M}(G)$.
Fixed a term order $<$ on $\mathcal{T}$, we have the following results:

Lemma / Definition 1.1.1 ([70, 79])**.** It holds:
$\mathcal{P} \cong I \oplus \mathbf{k}[\mathsf{N}(I)]$;
$\mathcal{P}/I \cong \mathbf{k}[\mathsf{N}(I)]$;
$\forall f \in \mathcal{P}, \exists! g := Can(f, I) = \sum_{\tau \in \mathsf{N}(I)} \gamma(f, \tau, <)\tau \in \mathbf{k}[\mathsf{N}(I)]$, called *canonical form* of f with respect to I, such that $f - g \in I$.

Definition 1.1.2 ([20])**.** Given a term order $\preceq$, a monomial basis for $A := \mathcal{P}/I(\mathbf{X})$, $[\tau_1], ..., [\tau_S]$, with $\tau_1 \preceq ... \preceq \tau_S$ is *minimal* w.r.t $\preceq$ if, for each monomial basis $[\tau'_1], ..., [\tau'_S]$, with $\tau'_1 \preceq ... \preceq \tau'_S$ it holds $\forall j = 1, ..., S,\ \tau_j \preceq \tau'_j$.

We will usually denote a monomial basis for a quotient algebra only with the terms, omitting the square brackets.

Definition 1.1.3 ([79])**.** A *Groebner basis* of I is a set $\mathcal{G} \subset I$ such that $\mathsf{T}(\mathcal{G}) = \mathsf{T}\{I\}$;
a *minimal Groebner basis* is a Groebner basis $\mathcal{H}$ such that do not exist divisibility relations among the leading terms of its members: $\mathsf{T}\{\mathcal{H}\} = \mathsf{G}(I)$;
the unique *reduced Groebner basis* of I is the set: $\mathcal{G}'(I) := \{\tau - Can(\tau, I) : \tau \in \mathsf{G}(I)\}$. Each member of the reduced Groebner basis has a leading term which does not divide any term of another member.

Unless otherwise specified, we consider the *lexicographic order* induced by $(x_0 <) x_1 < ... < x_n$, i.e:

$$(x_0^{\alpha_0})x_1^{\alpha_1} \cdots x_n^{\alpha_n} < (x_0^{\beta_0})x_1^{\beta_1} \cdots x_n^{\beta_n} \Leftrightarrow \exists j \,|\, \alpha_j < \beta_j,\ \alpha_i = \beta_i,\ \forall i > j.$$

This is a *term order*, that is a semigroup ordering such that 1 is lower than every variable or, equivalently, it is a *well ordering*.
If $\mathsf{N} = \{\tau_1, ..., \tau_m\}$ is an order ideal and $\tau_1 < ... < \tau_m$ w.r.t. lex, then also $\mathsf{N}' = \{\tau_1, ..., \tau_h\}$ is

an order ideal, $\forall h < m$.
A term order is called *degree compatible* if, for each $\tau_1, \tau_2 \in \mathcal{T}$,

$$deg(\tau_1) < deg(\tau_2) \Rightarrow \tau_1 < \tau_2.$$

Let $\mathbf{X} = \{P_1, ..., P_S\} \subset \mathbf{k}^n$ be a finite set of distinct points,

$$P_i := (a_{i1}, ..., a_{in}),\ i = 1, ..., S,$$

the *ideal of points* of $\mathbf{X}$ is

$$I(\mathbf{X}) := \{f \in \mathcal{P} :\ f(P_i) = 0,\ \forall i\}.$$

On the contrary, if $I \triangleleft \mathcal{P}$ is an ideal, we define its associated *variety* as

$$V(I) = \{P \in \mathbf{k}^n,\ f(P) = 0,\ \forall f \in \mathcal{P}\}.$$

For each $1 \leq m \leq n$, we define the projection maps as:

$$\pi_m : \mathbf{k}^n \to \mathbf{k}^m \qquad\qquad \pi^m : \mathbf{k}^n \to \mathbf{k}^{n-m+1}$$

$$(X_1, .., X_n) \mapsto (X_1, ..., X_m), \qquad\qquad (X_1, .., X_n) \mapsto (X_m, ..., X_n)$$

and, for $P \in \mathbf{k}^n,\ \mathbf{X} \subset \mathbf{k}^n$, let

$$\Pi_s(P, \mathbf{X}) := \{P_i \in \mathbf{X} | \pi_s(P_i) = \pi_s(P)\},$$

$$\Pi^s(P, \mathbf{X}) := \{P_i \in \mathbf{X} | \pi^s(P_i) = \pi^s(P)\},$$

extending in the obvious way the meanings of $\pi_s(\mathbf{d}), \pi^s(\mathbf{d}), \Pi_s(\mathbf{d}, D), \Pi^s(\mathbf{d}, D)$ to $\mathbf{d} \in \mathbb{N}^n$ and $D \subseteq \mathbb{N}^n$.
With the same notation π_m we indicate also

$$\pi_m : \mathcal{T} \cong \mathbb{N}^n \to \mathbb{N}^m \cong \mathcal{T}[m] \quad \text{such that} \quad x_1^{a_1} \cdots x_n^{a_n} \mapsto x_1^{a_1} \cdots x_m^{a_m}.$$

Consider and ideal $I \triangleleft \mathcal{P}$. We denote the set of polynomials in I with degree lesser or equal then d by $I(d) = I \cap \mathcal{P}(d)$. Such a set is a vector subspace of the vector space $\mathcal{P}(d)$.

Definition 1.1.4. Let $I \triangleleft \mathcal{P}$ be an ideal. The *affine Hilbert function* of I is the function

$$HF_I : \mathbb{N} \to \mathbb{N}$$

$$d \mapsto dim(\mathcal{P}(d)/I(d)).$$

For d sufficiently large, the affine Hilbert function of I can be written as

$$HF_I(d) = \sum_{i=0}^{l} b_i \binom{d}{l-i},$$

where l is the Krull dimension of $V(I)$,b_i are integers and b_0 is positive.

Definition 1.1.5. The polynomial which is equal to $HF_I(d)$ for d sufficiently large is called the *affine Hilbert polynomial* of I. It is denoted by $H_I(d)$.

We describe now the analogous concepts for the homogeneous case. Let $\mathcal{S}_d \subset \mathcal{S}$ be the set consisting of the homogeneous polynomials of total degree d and the polynomial 0, and $I_d = I \cap \mathcal{S}_d$ with $I \lhd S$ homogeneous ideal.

Definition 1.1.6. With the above notation, the *Hilbert function* of I is

$$^hHF_I : \mathbb{N} \to \mathbb{N}$$

$$d \mapsto dim(\mathcal{S}_d/I_d).$$

Given a homogeneous ideal $I \lhd S$, for d sufficiently large, we can write the Hilbert function as a polynomial, namely

$$^hHF_I(d) = \sum_{i=0}^{l} b_i \binom{d}{l-i},$$

i.e. the *Hilbert polynomial* of I, denoted by $^hH_I(d)$.
Finally, we recall the following definitions, which will be particularly useful in chapter 6.

Definition 1.1.7. Let $F = \{\tau_1, ..., \tau_s\} \subseteq \mathcal{T}$ be an ordered subset of terms, generating an ideal $J = (F)$. The module

$$Syz(F) = \{(g_1, ..., g_s) \in P^s, \sum_{i=1}^{s} g_i\tau_i = 0\}$$

is the *syzygy module* of F.
We denote an element in $Syz(F)$ by $(g_1, ..., g_s)$ and we call it *syzygy among* F.

Definition 1.1.8. The *S-polynomial* of two polynomials f and g w.r.t. a term ordering $<$, such that $Lc(f) = Lc(g) = 1$ is

$$S(f,g) := \frac{lcm(\mathsf{T}(f), \mathsf{T}(g))}{\mathsf{T}(f)} f - \frac{lcm(\mathsf{T}(f), \mathsf{T}(g))}{\mathsf{T}(g)} g$$

1.2 Groebner duality

In this section, we consider $\mathcal{P} = \mathbf{k}[x_1, ..., x_n]$ as a $\mathbf{k}$-vector space. In this perspective, we define the $\mathbf{k}$-functionals on $\mathcal{P}$.

Definition 1.2.1. A $\mathbf{k}$-functional l on $\mathcal{P}$ is a linear morphism $l : \mathcal{P} \to \mathbf{k}$, i.e. an element of the $\mathbf{k}$-vector space $\mathcal{P}^* := Hom_{\mathbf{k}}(\mathcal{P}, \mathbf{k})$.

We point out that

$$f \in \mathcal{P}, l \in \mathcal{P}^* \Rightarrow l(f) = \sum_{\tau \in \mathcal{T}} c(f, \tau) l(\tau).$$

We can equip $\mathcal{P}^*$ with a $\mathcal{P}$-modulo structure, defining $\forall l \in \mathcal{P}^*,\ f \in \mathcal{P}$

$$(l \cdot f)(g) := l(fg),\ \forall g \in \mathcal{P}.$$

Definition 1.2.2. Two sets $L := \{l_1, ..., l_s\} \subseteq \mathcal{P}^*$ and $q = \{q_1, ..., q_s\} \subseteq \mathcal{P}$ are called:

- *triangular* if $l_i(q_j) = 0,\ \forall i < j$;
- *biorthogonal* if $l_i(q_j) = \delta_{ij} = \begin{cases} 1,\ i = j \\ 0,\ i \neq j \end{cases}$

Given a $\mathbf{k}$-vector subspace $L \subseteq \mathcal{P}^*$ let

$$\mathfrak{P}(L) := \{g \in \mathcal{P} \,|\, l(g) = 0, \forall l \in L\}$$

and, for each $\mathbf{k}$-vector subspace $Q \subseteq \mathcal{P}$ let

$$\mathcal{L}(Q) = \{l \in \mathcal{P}^* \,|\, l(g) = 0,\ \forall g \in Q\}.$$

Definition 1.2.3. A subset of $\mathcal{P}^*$ is called *dual basis* of a $\mathbf{k}$-vector subspace $Q \subset \mathcal{P}$ if it is a basis of $\mathcal{L}(Q)$.

Lemma / Definition 1.2.4 ([1])**.** For each $\mathbf{k}$-vector subspace $Q, Q_1, Q_2 \subset \mathcal{P}$, $L, L_1, L_2 \subset \mathcal{P}^*$ it holds:

1. $Q \triangleleft \mathcal{P} \Rightarrow \mathcal{L}(Q)$ is a $\mathcal{P}$-module;
2. L is a $\mathcal{P}$-module $\Rightarrow \mathfrak{P}(L) \triangleleft \mathcal{P}$;
3. $Q_1 \subseteq Q_2 \Rightarrow \mathcal{L}(Q_1) \supseteq \mathcal{L}(Q_2)$;
4. $L_1 \subseteq L_2 \Rightarrow \mathfrak{P}(L_1) \supseteq \mathfrak{P}(L_2)$;

5. $\mathcal{L}(Q_1 \cap Q_2) \supset \mathcal{L}(Q_1) + \mathcal{L}(Q_2)$;
6. $\mathfrak{P}(L_1 \cap L_2) \supset \mathfrak{P}(L_1) + \mathfrak{P}(L_2)$;
7. $\mathcal{L}(Q_1 + Q_2) = \mathcal{L}(Q_1) \cap \mathcal{L}(Q_2)$;
8. $\mathfrak{P}(L_1 + L_2) = \mathfrak{P}(L_1) \cap \mathfrak{P}(L_2)$;
9. $Q = \mathfrak{P}(\mathcal{L}(Q))$;
10. $L \subset \mathcal{L}(\mathfrak{P}(L))$;
11. $dim_k(L) < \infty \Rightarrow L = \mathcal{L}(\mathfrak{P}(L))$.

An ideal has a finite dual basis $(L_1, ..., L_s)$ if and only if it is zerodimensional of degree s. $\mathfrak{P}$ and $\mathcal{L}$ define a *duality* between finite dimensional $\mathcal{P}$-modules of functionals and zerodimensional ideals.

Let $\mathbf{X} = \{P_1, ..., P_S\} \subset \mathbf{k}^n$ a finite set of points

$$P_i := (a_{i1}, ..., a_{in}),\ i = 1, ..., S.$$

For each i we denote by $l_i \in \mathcal{P}^*$ the linear functional consisting of the evaluation at P_i, i.e.

$$l_i(f) = ev_{P_i}(f) = f(a_{i1}, ..., a_{in}),\ \forall f(x_1, ..., x_n) \in \mathcal{P},$$

We can extend definition 1.2.2 in order to work with finite sets of distinct points.
If $\mathbf{X} = \{P_1, ..., P_S\}$ is such a set and $q = \{q_1, ..., q_S\} \subseteq \mathcal{P}$, we say that they are triangular (biorthogonal) if, letting $l_i :=$ evaluation at P_i, $\forall 1 \leq i \leq S$, q and $L := \{l_1, ..., l_S\}$ are triangular (biortogonal).
Then, we call

$$L(\mathbf{X}) := Span_{\mathbf{k}}(\{l_i,\ 1 \leq i \leq S\}) \subset \mathcal{P}^*;$$

which is dual to the ideal of points $I(\mathbf{X})$.
Now, we loosely base on [73], sketching the main properties of *differential operators*.
For each $i_1, ..., i_n \in \mathbb{N}$ define the differential operators

$$D(i_1, ..., i_n) : \mathcal{P} \to \mathcal{P}$$

given by

$$\frac{1}{i_1! \cdots i_n!} \frac{\partial^{i_1 + ... + i_n}}{\partial x_1^{i_1} \cdots \partial x_n^{i_n}}.$$

The summation $i_1 + ... + i_n$ is called *degree* of $D(i_1, ..., i_n)$.
By the natural isomorphism $\mathbb{N}^n \cong \mathcal{T}$ we indifferently use the notation $D(i_1, ..., i_n)$ and $D(\tau)$,

if $\tau = x_1^{i_1} \cdots x_n^{i_n} \in \mathcal{T}$.
We point out that $D(0, ..., 0) = D(1)$ is the identity.
Then, we denote $D := \{D(\tau) | \tau \in \mathcal{T}\}$ and by $Span_{\mathbf{k}}(D)$ the $\mathbf{k}$-vector space generated by D and we define the degree of an element in $Span_{\mathbf{k}}(D)$ as the maximal degree of the $D(\tau)$'s appearing in it.
For each $j = 1, ..., n$ we define $\sigma_{x_j} : D \to D \cup \{0\}$ the *antiderivative* w.r.t. x_j as

$$\sigma_{x_j}(D(i_1, ..., i_n)) := D(i_1, ..., i_j - 1, ..., i_n) \text{ if } i_j \geq 1$$

$$\sigma_{x_j}(D(i_1, ..., i_n)) := 0 \text{ if } i_j = 0$$

We use the notation $\sigma_{x_i x_j}$ for $\sigma_{x_i}\sigma_{x_j} = \sigma_{x_j}\sigma_{x_i} =: \sigma_{x_i x_j}$ and, for each $\tau \in \mathcal{T}$, defining $\sigma_{\tau x_j} = \sigma_{x_j}\sigma_\tau$, we have a map $\sigma_\tau : D \to D \cup \{0\}$, which can be extended to a $\mathbf{k}$-endomorphism of $Span_{\mathbf{k}}(D)$ still denoted by σ_τ.

We notice that

$$\forall \tau, \tau' \in \mathcal{T}, \sigma_\tau \sigma_{\tau'} = \sigma_{\tau\tau'}$$

and we point out that $\sigma_\tau D(\mu) \neq 0$ if and only if $\tau | \mu$.

Definition 1.2.5. A $\mathbf{k}$-vector subspace $V \subset Span_{\mathbf{k}}(D)$ is *closed* if the following conditions hold:

1. $dim_k V \leq \infty$;
2. $\forall \tau \in \mathcal{T}$, $\forall \partial \in V$, $\sigma_\tau(\partial) \in V$.

Let $P = (a_1, ..., a_n) \in \mathbf{k}^n$ and $\mathcal{M}(P) = (x_1 - a_1, ..., x_n - a_n) \lhd \mathcal{P}$ be the corresponding maximal ideal and $ev(P)$ the evaluation functional in P.
Each $\partial \in Span_{\mathbf{k}}(D)$ induces a functional $\partial(P) \in \mathcal{P}^*$ defined by $\partial(P)(f) = ev(P)(\partial f)$.

Proposition 1.2.6. $\forall f \in \mathcal{P}$, $\partial \in D$

$$\partial(x_k f) = x_k \partial(f) + \sigma_{x_k}(\partial)(f)$$

therefore

$$\partial(P)(x_j g) = a_j \partial(P)(g) + ev(P)(\sigma_{x_j}(\partial)(g)).$$

Proposition 1.2.7. Let $P \in \mathbf{k}^n$, $\Delta := \{\partial_1, ..., \partial_r\} \subset Span_{\mathbf{k}}(D)$; then the set

$$Q := \{f \in \mathcal{P} | \partial_i(P)(f) = 0,\ i = 1, ..., r\}$$

is an ideal if and only if Δ is closed.

Proposition 1.2.8. Let $P \in \mathbf{k}^n$, $\mathcal{M}(P)$ the corresponding maximal ideal and $V \subset Span_{\mathbf{k}}(D)$ a closed subspace; then

$$\mathcal{J}_P(V) := \{f \in \mathcal{P} | \partial(P)(f) = 0, \forall \partial \in V\}$$

is an $\mathcal{M}(P)$-primary ideal.

Proposition 1.2.9. There is a one to one correspondence between the $\mathcal{M}(P)$-primary ideals of $\mathcal{P}$ and the closed subspaces of $Span_{\mathbf{k}}(D)$.
More precisely, each $\mathcal{M}(P)$-primary ideal $\mathcal{Q}$ corresponds to a closed subspace

$$\Delta_P(\mathcal{Q}) := \{\partial | \partial(P)(f) = 0, \forall f \in \mathcal{Q}\},$$

while each closed subspace $V \subset Span_{\mathbf{k}}(D)$ corresponds to the $\mathcal{M}(P)$-primary ideal

$$\mathcal{J}_P(V) := \{f \in \mathcal{P} | \partial(P)(f) = 0, \forall \partial \in V\}.$$

Moreover,

$$\dim_k(\Delta_P(\mathcal{Q})) = \text{mult}(\mathcal{Q}) = deg(\mathcal{Q})\ e\ \text{mult}(\mathcal{J}_P(V)) = \dim_k(V).$$

Let $\mathcal{M} \triangleleft \mathcal{P}$ be a maximal ideal without zeroes in $\mathbf{k}^n$ and $\mathbf{Y} = \{P_1, P_2, ..., P_r\}$ its zeroes in $\overline{\mathbf{k}}^n$, where $\overline{\mathbf{k}}$ is the algebraic closure of $\mathbf{k}$. We call $\mathbf{k}_i$ the minimal algebraic field extension of $\mathbf{k}$, containing all the coordinates of P_i.

Proposition 1.2.10. Let $\mathcal{M} \triangleleft \mathcal{P}$ be a maximal ideal without zeroes in $\mathbf{k}^n$ and $\mathbf{Y} = \{P_1, P_2, ..., P_r\}$ its zeroes in $\overline{\mathbf{k}}^n$, where $\overline{\mathbf{k}}$ is the algebraic closure of $\mathbf{k}$. Then there is a one to one correspondence, between $\mathcal{M}$-primary ideals and the closed subspaces of $Span_{\mathbf{k}_1}(D)$.
Each $\mathcal{M}$-primary ideal $\mathcal{Q}$ corresponds to the closed subspace of $Span_{\mathbf{k}_1}(D)$

$$\Delta(\mathcal{Q}) = \{\partial | \partial(P_1)(f) = 0 \, \forall f \in \mathcal{Q}\}.$$

To each closed subspace $V \subset Span_{\mathbf{k}_1}(D)$ corresponds the $\mathcal{M}$-primary ideal

$$\mathcal{J}(V) = \{f \in P | \partial(P_1)(f) = 0, \forall \partial \in V\},$$

so that $\mathcal{Q} = \mathcal{J}(\Delta(\mathcal{Q}))$ and $V = \Delta(\mathcal{J}(V))$.

Theorem 1.2.11. Every 0-dimensional ideal $I \triangleleft \mathcal{P}$ is uniquely defined by a set of points $P_1, ..., P_r \in \overline{\mathbf{k}}^n$ ($\overline{\mathbf{k}}$ the algebraic closure of $\mathbf{k}$) which are not conjugate over $\mathbf{k}$ and, for any point $P_i = (a_{i1}, ..., a_{in})$ a closed subspace

$$\Delta_i = Span_{\mathbf{k}_i}(\partial_{i1}, ..., \partial_{in}) \subset Span_{\mathbf{k}_i}(D),$$

$\mathbf{k}_i = \mathbf{k}(a_{i1}, ..., a_{in})$ so that $f \in I$ if and only if $\forall i, j,\ \partial_{ij}(P)(f) = 0$.
For each i, let $\alpha_{i_1}, ..., \alpha_{i_{t_i}}$ a $\mathbf{k}$-basis of $\mathbf{k}_i$ so that $\forall i, j$ exist $\mathbf{k}$-functionals $L_{ijk} \in \mathcal{P}^*$, $k = 1, ..., t_i$ with $\partial_{ij}(P_i)(f) = \sum L_{ijk}(f)\alpha_{ik}$.

Then I is defined by $\{L_{ijk} | i = 1, ..., r,\ j = 1, ..., s_i,\ k = 1, ..., t_i\}$.

Now, following [2], we give a glimmer of Macaulay bases.
For each polynomial $f \in \mathcal{P}$ (or for each series f) we denote by $L(f)$ its *lowest degree non-zero homogeneous component*, whereas $ord(f) = deg(L(f))$ is its *order* or *underdegree*.
We fix an infinite set of indeterminates, labeled with the elements in $\mathcal{T}$, namely $Z = \{\zeta_\tau,\ \tau \in \mathcal{T}\}$ and we have naturally the rings $\mathbf{k}[\zeta_\tau]_{\tau\in\mathcal{T}}$ and $\mathbf{k}[[\zeta_\tau]]_{\tau\in\mathcal{T}}$.

Definition 1.2.12. A *dialytic equation* of an ideal $I \triangleleft \mathcal{P}$ is a linear combination

$$\sum_{\tau\in\mathcal{T}} a_\tau \zeta_\tau \in \mathbf{k}[\zeta_\tau]_{\tau\in\mathcal{T}}$$

such that

$$\sum_{\tau\in\mathcal{T}} a_\tau \tau \in I.$$

For each term $\nu \in \mathcal{T}$, the *ν-derivative* of $\sum_{\tau\in\mathcal{T}} a_\tau\zeta_\tau$ is the dialytic equation $\sum_{\tau\in\mathcal{T}} a_\tau\zeta_{\tau\nu}$, corresponding to

$$\sum_{\tau\in\mathcal{T}} a_\tau\tau\nu = \nu \sum_{\tau\in\mathcal{T}} a_\tau\tau \in I.$$

Definition 1.2.13. The *inverse functions* or *modular equations* of I are the equations of the form

$$\sum_{\tau\in\mathcal{T}} c_\tau\zeta_\tau \in \mathbf{k}[[\zeta_\tau]]_{\tau\in\mathcal{T}},$$

with $\sum_{\tau\in\mathcal{T}} c_\tau a_\tau = 0$, for each

$$\sum_{\tau\in\mathcal{T}} a_\tau\tau \in I.$$

We can naturally extend the notion of lowest degree component and order to dialytic equations and inverse functions and, for each inverse function $\sum_{\tau\in\mathcal{T}} c_\tau\zeta_\tau \in \mathbf{k}[[\zeta_\tau]]_{\tau\in\mathcal{T}}$, we can define a linear functional $\gamma \in \mathcal{P}^*$, namely the one associating the element c_τ to each τ. Following Macaulay's notation, we express these equations as Laurent series

$$\sum_{\tau\in\mathcal{T}} c_\tau\tau^{-1} = \sum_{(\alpha_1,...,\alpha_n)\in\mathbb{N}^n} c_{(\alpha_1,...,\alpha_n)} x_1^{-\alpha_1}\cdots x_n^{-\alpha_n} \in \mathbf{k}[[x_1^{-1}, ..., x_n^{-n}]].$$

The set of all Laurent series which are inverse functions of I is called *inverse system.*

Definition 1.2.14. An inverse function $\sum_{\tau\in\mathcal{T}} c_\tau \tau^{-1}$ for which exists $\gamma \in \mathbb{N}$ such that

$$deg(\tau) > \gamma \Rightarrow c_\tau = 0$$

is called *Noetherian equation*.

For each term $\tau \in \mathcal{T}$ we can define a functional

$$M(\tau) : \mathcal{P} \to k$$

$$f \mapsto c(f, \tau),$$

for each $f = \sum_{\tau\in\mathcal{T}} c(f,\tau)\tau \in \mathcal{P}$.
We denote by $\mathbb{M} = \{M(\tau),\, \tau \in \mathcal{T}\}$ the set containing all these functionals, whereas $Span_{\mathbf{k}}(\mathbb{M}) \subseteq \mathcal{P}^*$ is the **k**-vector space generated by $\mathbb{M}$.
Each semigroup ordering $<$ on $\mathcal{T}$ induces an ordering on $\mathbb{M}$:

$$M(\tau) \leq M(\omega) \Leftrightarrow \tau \leq \omega.$$

For each $l = \sum_{\tau\in\mathcal{T}} c(\tau, l)M(\tau) \in Span_{\mathbf{k}}(\mathbb{M})$, we define the *support* of l as

$$S(l) = \{\tau \in \mathcal{T},\, c(\tau, l) \neq 0\}.$$

If $f := \sum_{\tau\in\mathcal{T}} a_\tau\tau \in \mathcal{P}$ and $l := \sum_{\tau\in\mathcal{T}} c_\tau M(\tau) \in Span_{\mathbf{k}}(\mathbb{M})$ we have

$$l(f) = \sum_{\tau\in\mathcal{T}} a_\tau c_\tau = \sum_{\tau\in S(l)\cap S(f)} a_\tau c_\tau,$$

so $Span_{\mathbf{k}}(\mathbb{M})$ is the set of all the Noetherian equations.
For each $\Lambda \subset Span_{\mathbf{k}}(\mathbb{M})$ and for each **k**-vector subspace $\mathfrak{P} \subset \mathcal{P}$ we denote

$$\mathcal{I}(\Lambda) := \{f \in \mathcal{P} :\, l(f) = 0,\, \forall\lambda \in \Lambda\};$$

$$\mathcal{M}(\mathfrak{P}) := \{l \in Span_{\mathbf{k}}(\mathbb{M}),\, l(f) = 0,\, \forall f \in \mathfrak{P}\}.$$

In analogy with the antiderivatives for elements of $\mathcal{T}$, for each $j \in \{1, ..., n\}$, given $M \in Span_{\mathbf{k}}(\mathbb{M})$ we define for each $\tau \in \mathcal{T}$

$$\sigma_j(M(\tau)) := \begin{cases} M(\omega) \text{ if } \tau = x_j\omega \\ 0 \text{ if } x_j \nmid \tau \end{cases}$$

Since for each i, j $\sigma_i\sigma_j = \sigma_j\sigma_i$ we can define inductively $\sigma_\tau \in End_k(Span_{\mathbf{k}}(\mathbb{M}))$, for each $\tau \in \mathcal{T}$, $\sigma_{x_j\tau} := \sigma_{x_j}\sigma_\tau$ so that, for all $\tau', \omega \in \mathcal{T}$

$$\sigma_{\tau'}(M(\omega)) := \begin{cases} M(\nu) \text{ if } \omega = \tau'\nu \\ 0 \text{ if } \tau' \nmid \omega \end{cases}$$

We can extend the definition to polynomials: $\forall f = \sum_i c_i\tau_i \in \mathcal{P}$,

$$\sigma_f(l) := \sum_i c_i \sigma_{\tau_i}(l)$$

and we equip $Span_{\mathbf{k}}(\mathbb{M})$ with a $\mathcal{P}$-module structure, letting

$$lf := \sigma_f(l),\ \forall f \in \mathcal{P},\ \forall l \in Span_{\mathbf{k}}(\mathbb{M}).$$

Definition 1.2.15. A $\mathbf{k}$-vector subspace Λ of $Span_{\mathbf{k}}(\mathbb{M})$ is called x_j-*stable* if $\sigma_i(l) \in \Lambda, \forall l \in \Lambda$ and *stable* if $\sigma_f(l) \in \Lambda$, $\forall l \in \Lambda$ and $f \in \mathcal{P}$.

If $l := \sum_i c_i M(\tau_i) \in Span_{\mathbf{k}}(\mathbb{M})$, $c_i \in \mathbf{k} \setminus \{0\}$, $\tau_i \in \mathcal{T}$, $\tau_1 < \tau_2 < ... < \tau_i < ...$, we denote by $T_<(l) = \tau_1$ the *leading term* of l and for $\Lambda \subset Span_{\mathbf{k}}(\mathbb{M})$ $T_<\{\Lambda\} := \{T_<(l),\ l \in \Lambda\}$, $N_<\{\Lambda\} := \mathcal{T} \setminus T_<\{\Lambda\}$.

Definition 1.2.16. Referring to definition 1.2.14 and to the comments above, a basis $\{l_1, ..., l_i, ...\}$ of a stable vector subspace $\Lambda \subset Span_{\mathbf{k}}(\mathbb{M})$ is a *Macaulay basis* of Λ w.r.t. an inf-limited ordering $<$ if

1. $T\{\Lambda\} = \{T(l_i)\} \subseteq \mathcal{T}$ is an order ideal;
2. $l_i = M(T(l_i)) + \sum_{\nu \in N(\Lambda)} \xi(\nu, T(l_i))M(\nu)$, for each i and suitable $\xi(\nu, T(l_i)) \in k$.

We conclude this section defining a special kind of ideals, called *Cerlienco-Mureddu ideals*. For each zerodimensional ideal $I \lhd \mathcal{P}$, we set $\mathbf{X} = V(I)$; $\forall P = (a_1, ..., a_n) \in \mathbf{X}$ we define

$$\lambda_P : \mathcal{P} \to \mathcal{P}$$

$$x_i \mapsto x_i + a_i,\ i = 1, ..., n,$$

$\mathcal{M}_P = (x_1 - a_1, ..., x_n - a_n)$ and $\mathcal{Q}_P$ the $\mathcal{M}_P$-primary component of I.
We define $\Lambda_P := \mathcal{M}(\lambda_P(\mathcal{Q}_P)) \subset Span_{\mathbf{k}}(\mathbb{M})$ and $\{\lambda_{\nu P} := l(\nu) : \nu \in \mathsf{N}_<(\lambda_P(\mathcal{Q}_P))\}$ the Macaulay basis of Λ_P.
We suppose it ordered so that each vector subspace $L_\sigma := Span_{\mathbf{k}}(\{l_{\nu_1}, ..., l_{\nu_\sigma}\})$ is a $\mathcal{P}$-module and we set

$$\mathbb{L} := \{\lambda_1, ..., \lambda_s\} = \{l_{\nu P}\lambda_P : \nu \in \mathsf{N}_<(\lambda_P(\mathcal{Q}_P)),\ P \in \mathbf{X}\},$$

ordered so that each vector subspace $L_\sigma := Span_{\mathbf{k}}(\{l_1, ..., l_\sigma\})$ is a $\mathcal{P}$-module.
Then, we set $\mathbf{Y} = \{Y_1, ..., Y_s\}\{(P, \nu) \in \mathsf{N}_<(\lambda_P(\mathcal{Q}_P)),\ P \in \mathbf{X}\}$ enumerated so that

$$Y_j = (P, \nu) \Leftrightarrow \lambda_j = l_{\nu P}\lambda_P.$$

Following [79], we suppose each $\lambda_P(\mathcal{Q}_P)$ to be a monomial ideal.

Definition 1.2.17. With the previous notation, the ordered sets $\mathbb{L}$ and $\mathbf{Y}$ are a *Macaulay representation* and a *Cerlienco-Mureddu skeleton* of $I := \mathfrak{P}(\mathbb{L})$; each $\lambda = l_{\nu P}\lambda_P$ is a *Cerlienco-Mureddu functional* and each $Y = (P, \nu) \in \mathbf{Y}$ a *Cerlienco-Mureddu card*.
Moreover, if $\forall \lambda = l_{\nu P}\lambda_P \in \mathbb{L}$, $\lambda = M(\lambda) = M(\nu)\lambda_P$ then I is a *Cerlienco-Mureddu ideal*.

1.3 Graphs, trees, forests.

Here we recall some basic notions of Graph Theory. For more details see [45].

Definition 1.3.1. A *graph* G is the datum of:

- a nonempty set $V(G)$ whose elements are called *vertices* or *nodes*;
- a set of non ordered couples of distinct vertices $E(G)$ whose elements are called *edges*.

We summarize some terminology of Graph Theory

Notation 1.3.2. The *degree* $deg(a)$ of a given vertex $a \in V(G)$ is the number of edges incident with a.
A *subgraph* of a given G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.
A *walk* in a graph G is a sequence $a_0, e_1, a_1, ..., e_h, a_h, a_0, ..., a_h \in V(G)$ and $e_1, ..., e_h \in E(G)$, such that a_{j-1}, a_j are connected by e_j, $j = 1, ..., h$.
A *path* is a walk whose set of vertices does not contain repeated elements; a *cycle* or *circuit* is a *closed walk* i.e. a walk such that $a_0 = a_h$.
A graph G is:

- *connected* if for any couple of vertices there exists a path joining them;
- *acyclic* or *forest* if it does not contain any cycle;
- a *tree* if it is acyclic and connected (any subgraph of a tree is also a tree). All the trees of more than one vertex contain at least two vertices of degree 1, called *leaves*.

To each graph G can be associated a picture consisting of points (corresponding to the nodes of $V(G)$) and segments (corresponding to the edges of $E(G)$).
In particular, for each drawing of a given tree, the topmost node is called *root* of the tree. A *rooted tree* is a tree with a conspicuous root.
Fixed a root, we can read the elements of a tree from the root to the leaves.
The *level* of a node in a tree is its distance from the root. In particular, the root is at level 0.
The *height* of a tree is the maximal level of its nodes.
In a couple of nodes connected by an edge (so that their levels differ by one) the node of lowest level is called *father* and the other one is called *child*. In a similar way, we speak of *ancestors* and *descendants* for connected nodes whose levels differ for more than one.

Definition 1.3.3. A *trie* is a rooted tree such that each edge is labeled by an element of a fixed alphabet.

1.4 Points, terms and towers.

In this section, we introduce a simple way to represent points and terms. It will be very useful, especially while studying the combinatorial methods to compute the Groebner escalier associated to the ideal of a finite set of distinct points. We will exploit the natural isomorphism $\mathcal{T} \leftrightarrow \mathbb{N}^n$, starting with the case of $n = 2$ and then generalizing to an arbitrary n.
Given a set $\mathbf{X}' = \{P'_1, ..., P'_S\} \subset \mathbf{k}^2$ let r be the number of distinct prime coordinates of the P_i's, we group the points w.r.t. their first coordinates, obtaining r subset $\mathbf{X}'_1 = \{P_{1,1}, ..., P_{1,l_1}\}, ..., \mathbf{X}'_r = \{P_{r,1}, ..., P_{r,l_r}\}$.
Each point $P_{i,j} = (a_{1,i,j}, a_{2,i,j}) \in \mathbf{X}'$ is represented in the plane as a rectangle, labeled with the couple $(a_{1,i,j}, a_{2,i,j})$. If $P_{i,j}, P_{k,l}$ belong to the same $\mathbf{X}'_h \subset \mathbf{X}'$, their corresponding rectangles are superimposed and the rectangle on the bottom is the one corresponding to the point appearing first in $\mathbf{X}'_h$, so each $\mathbf{X}'_h$ is said corresponding to a *tower* in the plane.

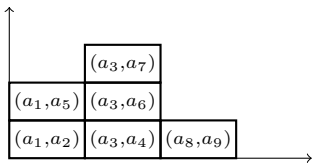


Figure 1.1: Tower structure in the plane (1).

The first tower has to be drawn so that the left side lies on the x_2-axis and all the subse-

quent towers have the left side lying on the right side of the previous one, as shown in the picture above.
Each rectangle in the tower is associated to a couple in $\mathbb{N}^2$, representing its *position*, in the following way

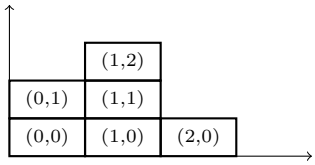


Figure 1.2: Tower structure in the plane (2).

Consider the isomorphism $\mathcal{T} \to \mathbb{N}^2$, sending a term $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \in \mathcal{T}$ to the point $(\alpha_1, \alpha_2) \in \mathbb{N}^2$. We can naturally associate to each point in the tower the term identified by its position. For the picture above, we get $\{1, x_1, x_1^2, x_2, x_1x_2, x_1x_2^2\}$.

Example 1.4.1. Let $\mathbf{X}' = \{(0,0), (1,1), (0,1), (1,2), (1,3)\}$.
Grouping the points w.r.t. their first coordinates we get $\mathbf{X'_0} = \{(0,0), (0,1)\}$
$\mathbf{X'_1} = \{(1,1), (1,2), (1,3)\}$.
and we cand draw the towers as in picture 1.3

Figure 1.3: The tower structure of $\mathbf{X}'$: points.

Identifying each term $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \in \mathbf{k}[x_1, x_2]$ with the point $(\alpha_1, \alpha_2) \in \mathbb{N}^2$ we can also draw the picture with terms below where points and terms are related by their position.

Let us see another example.

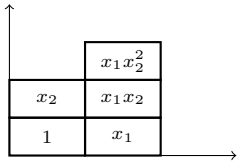
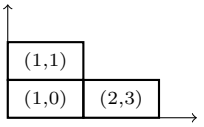
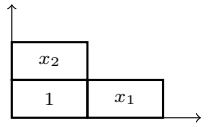


Figure 1.4: The tower structure of $\mathbf{X}'$: terms.

Example 1.4.2. For the set $\mathbf{X}' = \{(1,0),(2,3),(1,1)\}$ we get





Consider first example 1.4.1.
A Groebner basis (actually the reduced one, computed here using Singular, [30]) of $I(\mathbf{X}')$ w.r.t. lex induced by $x_1 < x_2$ is $\{x_1^2-x_1, x_1x_2^2-x_2^2-x_1x_2+x_2, x_2^3-2x_1x_2^2-4x_2^2+x_1^2x_2+7x_1x_2+3x_2-3x_1^2-3x_1\}$ and so the lexicographical Groebner escalier is $\mathsf{N} = \{1, x_1, x_2, x_1x_2, x_2^2\}$. Such a set *does not coincide* with the one identified by the towers we drew.
For 1.4.2, the situation is different. The reduced Groebner basis is $\{x_1^2-3x_1+2, x_1x_2-x_2-3x_1+3, x_2^2-x_2-6x_1+6\}$ and then the Groebner escalier is $\mathsf{N} = \{1, x_1, x_2\}$, coinciding with the one identified by the towers. If, in example 1.4.1 we shift to the right the point $(1,3)$, we obtain again a picture with towers but we have the coincidence as in 1.4.2. In the case $n = 2$, such a shifting can be avoided by *reordering the towers in decreasing order by height*. An explanation of this fact is given in chapter 2, especially in remark 2.2.8.
If the picture with towers of a set $\mathbf{X}'$ leads to the Groebner escalier of $I(\mathbf{X}')$, we call it *tower structure* of $\mathbf{X}'$. It is *mixed* if one or more shifts have been performed in order to obtain a representation of the Groebner escalier, *unmixed* otherwise.
Associating a tower structure to $\mathbf{X}'$, we notice that the horizontal lines represent the powers of x_1 appearing in terms with a fixed exponent of x_2.
It means that, if we take a term $\tau = x_1^{\alpha_1}x_2^{\alpha_2}$, all the other terms appearing in the horizontal line which contains τ are of the form $\sigma = x_1^{\beta_1}x_2^{\alpha_2}$.
Browsing these rows ordinately from the bottom to the top we associate to each one of them a power of x_2: more precisely to the lowest one x_2^0, to the one lying above x_2^1, and so on.
We call these horizontal lines x_2-*ranges*, while the x_1-ranges are the single rectangles. We will give a formal definition of range in chapter 5, while introducing the Bar Code structure.

Notice that also the exponents of x_1 are ordered if we read each line from left to right.
Now, given S points, we have associated them S terms. We consider the x_2-ranges, increasingly ordered with respect to the exponent of x_2 identifying them. Let $r_{2,0}, ..., r_{2,j}$ their cardinalities.
The terms of the x_2-range corresponding to x_2^0 are numbered from 1 to $r_{2,0}$, the ones of the x_2-range corresponding to x_2^1 from $r_{2,0}+1$ and $r_{2,0}+r_{2,1}$ and so on.
We can see an example of such a reordering in picture 1.5[2].

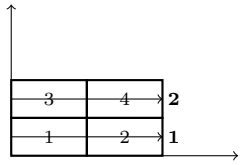


Figure 1.5: Reordering of ranges in 2 variables.

All these definitions can be generalized to the case of 3 or more variables.
We deal then with a set $\mathbf{X}'' = \{P_1'', ..., P_L''\} \subset \mathbf{k}^3$, constructing the towers similarly.

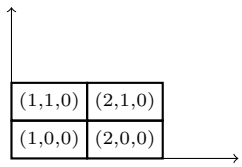
1. We draw the tower picture of $\mathbf{X}' := \pi_2(\mathbf{X})$. For each couple $(a_1, a_2) \in \mathbf{X}'$, label the rectangle corresponding to it with *one* of the points in the fiber $\pi_2^{-1}(a_1, a_2)$, say (a_1, a_2, a_3).

2. Since $\pi_2^{-1}(a_1, a_2)$ may contain more than one point, draw the rectangles corresponding to the elements of $\pi_2^{-1}(a_1, a_2) \setminus \{(a_1, a_2, a_3)\}$ over (a_1, a_2, a_3) along the x_3 direction.

Example 1.4.3. Consider the set

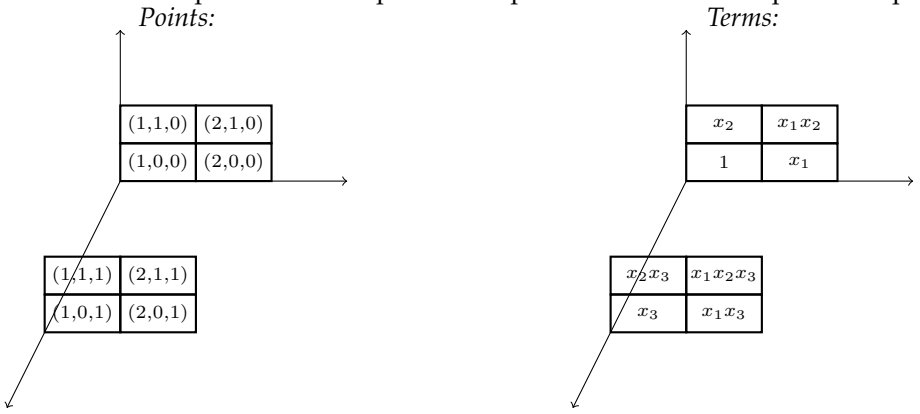
$$\mathbf{X}'' = \{(1,0,0), (1,0,1), (2,0,0), (1,1,0), (2,0,1), (1,1,1), (2,1,0), (2,1,1)\} \subseteq \mathbf{k}^3.$$

We have $\pi_2(\mathbf{X}'') = \{(1,0), (2,0), (1,1), (2,1)\}$:



[2]The x_1-ranges have been numbered using normal font numbers, while the x_2-ranges have been numbered using boldface numbers.

Since $\pi_2^{-1}(1,0) = \{(1,0,0),(1,0,1)\}$, $\pi_2^{-1}(2,0) = \{(2,0,0),(2,0,1)\}$, $\pi_2^{-1}(1,1) = \{(1,1,0),(1,1,1)\}$ and $\pi_2^{-1}(2,1) = \{(2,1,0),(2,1,1)\}$, we get the picture on the left. We display on the right the terms whose exponents' lists represent the positions in which the points are placed.



A Groebner basis of $I(\mathbf{X}'')$ w.r.t. lex induced by $x_1 < x_2 < x_3$ is $\{x_1^2-3x_1+2, x_2^2-x_2, x_3^2-x_3\}$ and so the corresponding Groebner escalier is $\{x_1x_2x_3, x_2x_3, x_1x_3, x_3, x_1x_2, x_2, x_1, 1\}$, which is an order ideal and it is exactly the set of terms characterized by the tower picture, which turns out to be an unmixed structure for $\mathbf{X}''$.

Consider the x_3-ranges, increasingly ordered with respect to the exponent of x_3 indentifying them and let $r_{3,0}, ..., r_{3,h}$ their cardinalities.
We number from 1 to $r_{3,0}$ the terms of the form $x_1^i x_2^j x_3^0$, according to the rule stated above for the case of two variables.
Then, we number from $r_{3,0}+1$ to $r_{3,0}+r_{3,1}$ the terms of the form $x_1^i x_2^j x_3^1$, according to the rule stated above for the case of two variables and so on.
Notice that if a term $\tau = x_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}$, belongs to a certain x_3-range, all the other terms of the same x_3-range are of the form $\sigma = x_1^{\beta_1}x_2^{\beta_2}x_3^{\alpha_3}$.
The following picture 1.6 represents an example of the reordering rule.

We numbered the 8 x_1-ranges with the normal font, the 4 x_2-ranges in boldface and the 2 x_3-ranges with the gothic font.
One can repeat all the construction (obtaining analogous mixed and unmixed tower structures) in the same way, applying it to any finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\} \subseteq \mathbf{k}^n$, $n > 3$ and generalize the idea of range.

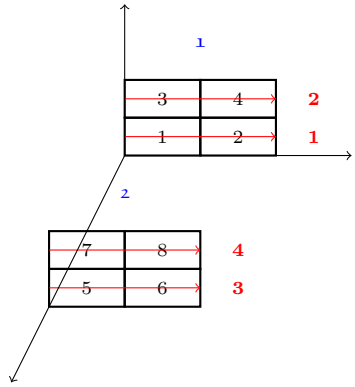


Figure 1.6: Reordering of ranges in 3 variables.

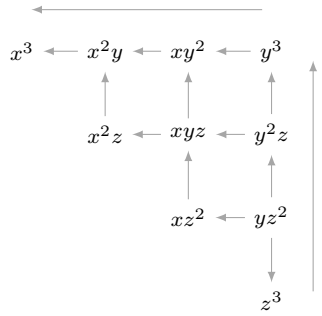
1.5 Graphical representation of terms in a small number of variables.

In this section we show how to represent graphically terms of degree r in 3, 4 or 5 variables. We will construct some diagrams, developed by M. G. Marinari and L. Ramella in [75] in order to draw strongly stable ideals.
Consider first the case of terms of degree r, belonging to the polynomial ring $\mathbf{k}[x, y, z]$ in 3 variables, ordered as $x < y < z$.
First of all, we draw on the bottom right the maximal variable (namely z), raised to the power r. Then, we construct a diagram, drawing the other terms, according to the rules below.

$\uparrow$: the exponent of z decreases by one, while the exponent of y increases by one;

$\leftarrow$: the exponent of y decreases by one, while the exponent of x increases by one.

Example 1.5.1. According to the rules $\uparrow$, $\leftarrow$, the diagram representing the 10 terms of degree 3 in three variables is:



As a matter of fact, every time we move up in the diagram above, the exponents of the variable z decrease, in favour of the powers of y.
In the same way, each step to the left means decreasing the exponent of y, making the one of x increase.

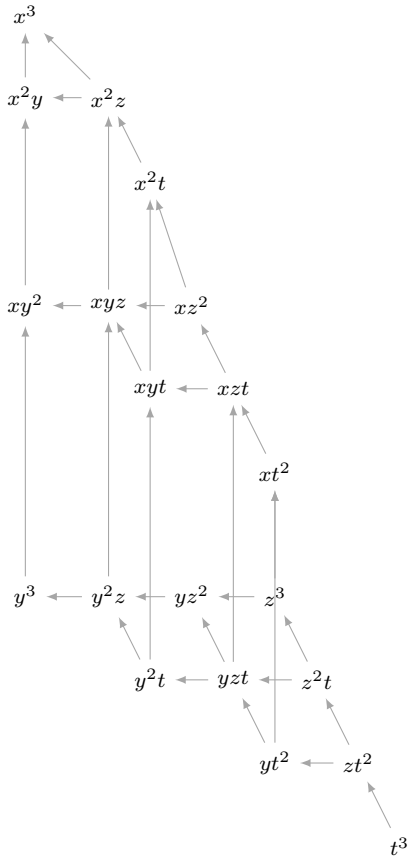
Suppose now to have one variable more, namely consider the polynomial ring $\mathbf{k}[x, y, z, t]$ with $x < y < z < t$.
We start again drawing on the bottom right the maximal term w.r.t. the lexicographical order, namely t^r. Then we extend the rules $\uparrow$, $\leftarrow$ explained for three variables as

$\nwarrow$: the exponent of t decreases by one, while the exponent of z increases by one;

$\leftarrow$: the exponent of z decreases by one, while the exponent of y increases by one;

$\uparrow$: the exponent of y decreases by one, while the exponent of x increases by one.

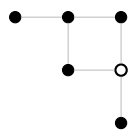
Example 1.5.2. The diagram representing the 20 terms of degree 3 in 4 variables is



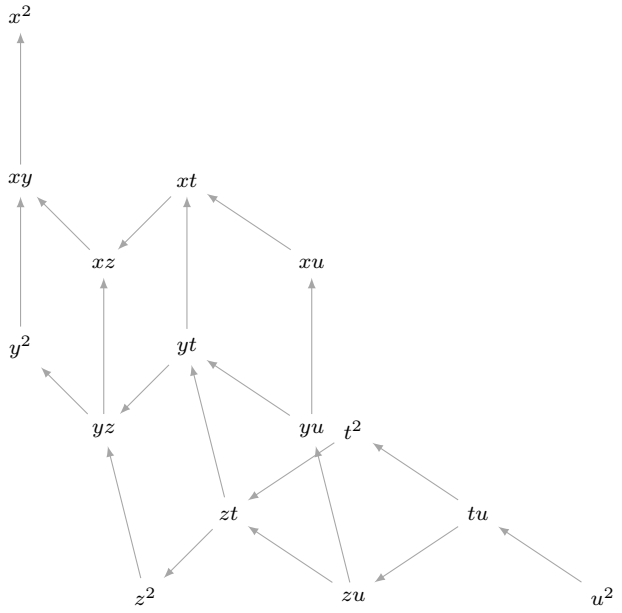
As in the case of three variables, the picture above follows the rules $\nwarrow$, $\uparrow$, $\leftarrow$.

For brevity's sake, we can also draw the diagram without specify the terms and substituting them with bullets. This method can be very useful in order to display the terms of a certain degree r, distinguishing the ones contained in a certain ideal and the ones belonging to the Groebner escalier.
For this purpose, we will use black bullets for the terms in the ideal and white bullets for the terms belonging to the associated Groebner escalier.

Example 1.5.3. Consider the ideal $I = (x, z^2, y^2) \lhd \mathbf{k}[x, y, z]$. At degree 2 we will have:



In chapter 5, we will introduce a new graphical representation for terms, allowing to increase *ad libitum* the number of variables.
We display here also a diagram in five variables



1.6 Moeller algorithm.

In 1982, Buchberger and Moeller ([12]) proposed an algorithm that, given a zerodimensional ideal I defined by s functionals $l_1, ..., l_s$ and a term order $<$, computes a Groebner basis and a triangular sequence $q_1, ..., q_s$ for a permutation $l_{\sigma(1)}, ..., l_{\sigma(s)}$ of the given functionals.
Many different versions of Moeller algorithm have been deeply studied by M.G. Marinari, H.M. Moeller and T. Mora in [73]. Here we briefly sketch two of them[3].
The first version is iterative on terms and it computes the reduced Groebner basis $\mathcal{G}$ and a triangular sequence $\mathbf{q}$. This version generalizes the original Buchberger-Moeller algorithm, for the case in which functionals are evaluations at a point.
The elements of the Groebner escalier and of the reduced Groebner basis are contained in two lists, which are updated in each iterative step, until each element of $\mathcal{T}$ is in $\mathsf{N}(I)$ or in $\mathsf{T}(\mathcal{G})$. At each step, the algorithm finds the minimal term τ not already settled in $\mathsf{N}(I)$ or in $\mathsf{T}(\mathcal{G})$ and computes $vect(\tau)$, the vector of evaluations of τ at the functionals. If $vect(\tau)$ is linearly dependent w.r.t. $\{vect(\sigma),\ \sigma \in \mathsf{N}(I)\}$ then a new element is added to $\mathcal{G}$; otherwise, we update the list $\mathbf{q}$.

Remark 1.6.1. We point out that Moeller algorithm is independent from the given term order $<$.

The algorithm leans on the subroutine GaussRed, which performs Gaussian reduction.

Algorithm 1 Gaussian reduction.

1: **procedure** GAUSSRED$(p, v, q_1, ..., q_r, vect(1), ..., vect(r)) \to p, v$
2: **for** $i = 1, ..., r$ **do**
3: $v = v - l'_i(p)vect(i)$
4: $p = p - l'_i(p)q_i$
5: **end for**
6: **end procedure**

The second version is iterative on functionals. At each step the Groebner escalier, the triangular sequence and the reduced Groebner basis are computed.

[3]There are also versions computing the Border basis [70].

Algorithm 2 Moeller algorithm 1.

procedure MOEL1($l_1, ..., l_s$) $\to \mathcal{G}, \mathbf{q}$
  $\mathcal{G} = \emptyset$
  $List = \{1\}$ ▷ List contains terms ordered w.r.t. $<$. Repeated elements are not allowed. Anyway the algorithm takes track of the number of times a repeated term would be inserted there.
  $\mathsf{N} = \emptyset$
  $r = 0$ ▷ N contains the Groebner escalier, while $r = |\mathsf{N}|$.
  while $List \neq \emptyset$ **do**
    $\tau := Min(List, <)$
    $List = List \setminus \{\tau\}$.
    if $\tau \notin \mathsf{T}(\mathcal{G})$ **then**
      $v = (l_1(\tau), ..., l_s(\tau))$
      $(p, v) = \text{GaussRed}(\tau, v, q_1, ..., q_r)$
      if $v = 0$ **then**
        $\mathcal{G} = \mathcal{G} \cup \{p\}$
      else
        $r++$
        $j = \min(i,\ l_i(p) \neq 0)$
        $l'_r = l_j$
        $vect(r) = l_j(p)^{-1} v$
        $q_r = l_j(p)^{-1} p$
        $\mathsf{N} = \mathsf{N} \cup \{\tau\}$
        $List = List \cup \{x_j \tau,\ \forall j\}$
      end if
    end if
  end while
end procedure

Algorithm 3 Moeller algorithm 2.

procedure MOEL2($l_1, ..., l_s$) $\to \mathcal{G}, \mathbf{q}$
  $\mathcal{G} = \{1\}$
  $v(1) = (l_1(1), ..., l_s(1))$
  for $r = 1...s$ **do**
    $\tau = \min\{\mathsf{T}(f),\ f \in \mathcal{G},\ l_r(f) \neq 0\}$
    let $f \in \mathcal{G}$, with $\mathsf{T}(f) = \tau$
    $\mathcal{G} = \mathcal{G} \setminus \{f\}$
    $q_r = l_r^{-1}(f)f$
    $vect(r) = l_r^{-1}(f)v(f)$
    for each $f \in \mathcal{G}$ s.t. $\mathsf{T}(f) > \tau$ **do**
      $f = f - l_r(f)q_r$
      $v(f) = v(f) - l_r(f)vect(r)$
    end for
    for $i = 1, ..., n$ **do**
      if $x_i\tau \notin (\mathsf{T}(\mathcal{G}))$ **then**
        $v = (l_1(x_i\tau), ..., l_s(x_i\tau))$
        $(p, v(p)) = \text{GaussRed}(x_i\tau, q_1, ..., q_r)$
        $\mathcal{G} = \mathcal{G} \cup \{p\}$
      end if
    end for
  end for
end procedure

In [73] is also proved that the computational complexity of the algorithms above is the same, in term of operations in k.
More precisely, denote by

- n the number of variables;
- s the number of functionals;
- $g = |\mathsf{G}|$;
- f the cost of functional evaluation.

The latter is actually a distributed cost: csf is the number of operations needed in order to evaluate s functionals at c terms.

More precisely:

- $f = 1$ if the functionals are evaluations at rational points;
- $f \leq s$ if the functionals are evaluations at algebraic points or evaluations of differential conditions at rational points[4].
- $f \leq 2ns$ if functionals include coefficients of canonical forms under a change of coordinates
- $f \leq s^2$ if functionals are evaluations at rational points with multiplicity conditions given by differential conditions.

Proposition 1.6.2. Both the algorithm have complexity

$$\frac{1}{2}s^3 + s^2 g + fs(s+g) \leq O(ns^3 + fns^2).$$

[4]In this case, a preprocessing is needed and it is polynomial in a natural measure for the input.

Part II

Combinatorics on the Groebner escalier.

CHAPTER 2

Combinatiorial methods for the Groebner escalier.

2.1 Introduction.

In this chapter, we summarize all the different methods to compute the Groebner escalier of a zerodimensional radical ideal $I(\mathbf{X}) \triangleleft \mathbf{k}[x_1, ..., x_n]$, the ideal of a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\} \subset \mathbf{k}^n$, where $\mathbf{k}$ is an arbitrary field.
These methods arose from the need to compute the Groebner escalier without passing through the Groebner basis computation, which can often be long and complicated.
Among these methods, we recall:

1. Cerlienco-Mureddu Correspondence;
2. Gao-Rodrigues-Stroomer method (and Lederer's variation);
3. the Lex Game.

The first one is iterative, while the others, requiring a preprocessing on the input points, drop out iterativity in favor of speed.
The most important methods are Cerlienco-Mureddu Correspondence and the Lex Game.
In chapters 4 and 5 we will see how to develop two algorithms which solve the same problem and are linked to the Lex Game, though having different aims.
The first one (providing an ordering both on the Groebner escalier and on $\mathbf{X}$) is an *interpolation oriented* algorithm as it is aimed to simplify the interpolation part. The second one is halfway between of $1.$ and $3.$, taking advantage of the main features of both of them.

2.2 Cerlienco-Mureddu Correspondence.

Cerlienco-Mureddu Correspondence is the very first algorithm that, given a finite set of distinct points $\mathbf{X}$, computes the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$ associated to $I(\mathbf{X})$ (without passing through a Groebner basis of $I(\mathbf{X})$) .
It dates back to the early nineties, with the articles [20, 21] where it is also generalized to the case of multiple points, using functionals[1].
Cerlienco-Mureddu face first (see [20], p. $1, 2$) the following problems:

(1) given $\mathbf{X}$ and S distinct values $\chi_1, ..., \chi_S$ find $p \in \mathbf{k}[x_1, ..., x_n]$ such that $p(P_i) = \chi_i$, $i = 1, ..., S$;

(2) analogous to (1) but knowing also the values of some partial derivatives (possibly different ones for each point) at P_i.

In order to have existence and uniqueness for the solution of (1), they force p to be of the form $a_1\tau_1 + ... + a_S\tau_S$, where $a_i \in k$, $i = 1, ..., S$ and $\tau_1, ..., \tau_S$ are terms such that their equivalence classes modulo $I(\mathbf{X})$ form a (monomial) basis for the quotient algebra $A := \mathbf{k}[x_1, ..., x_n]/I(\mathbf{X})$. Moreover, they require the monomial basis to be minimal (see definition 1.1.2) w.r.t. the given term order $<$
The solution of problem (1) is immediate if a monomial basis is known. In [20] three different solutions for this subproblem are proposed ([20] p.2):

1. look for a nonzero order S minor of a suitable matrix;

2. start from a system of generators for $I(\mathbf{X})$ and use the Groebner bases theory;

[1] For example evaluations of polynomials and their derivatives at points.

3. use a *purely combinatorial algorithm* (the one we call Cerlienco-Mureddu correspondence). giving a minimal monomial basis for the quotient, w.r.t lex induced by $x_1 \prec .... \prec x_n$) directly from $\mathbf{X}$.

Remark 2.2.1. We point out that the Groebner escalier associated to a zerodimensional ideal I is also provided by Moeller algorithm [1, 12, 73].

2.2.1 The elementary ideal and problem (1).

In the first part of [20], Cerlienco-Mureddu describe a way to compute a system of generators for a zerodimensional radical ideal $I(\mathbf{X})$, given $\mathbf{X} = \{P_1, ..., P_S\}$. We only sketch it and show a very simple example.

Definition 2.2.2. An ideal $I \lhd \mathbf{k}[x_1, ..., x_n]$ is called *elementary* if it is generated exactly by n polynomials, each one containing only one variable, i.e.

$$I = (\gamma_1(x_1), ..., \gamma_n(x_n)).$$

Clearly this set of polynomials is also its reduced Groebner basis w.r.t. any term order. Take then $\mathbf{X}$ and perform the following steps.

1. Associate to it an elementary ideal I':
   - take the supset $\mathbf{X}'$ of $\mathbf{X}$ consisting of the points $P = (a_1, ..., a_n) \in \mathbf{k}^n$ such that for each $1 \leq j \leq n$, a_j is the j-th coordinate of some point of $\mathbf{X}$ [2];
   - $I' = I(\mathbf{X}')$ is an elementary ideal, say $I' = (\gamma_1, ..., \gamma_n)$, where $\gamma_j \in \mathbf{k}[x_j]$, $deg(\gamma_j) = h_j$ are such that $\gamma_j(a) = 0$ if and only if a is the j-th coordinate of at least a point in $\mathbf{X}$.
2. Observe that the Groebner escalier turns out to be
$$\mathsf{N}_<(\mathbf{X}') = \{x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \, | \, 0 \leq \alpha_j \leq h_j - 1\} = \{x^{\alpha^{(1)}}, ..., x^{\alpha^{(n)}}\};$$
3. Let $\mathcal{H}$ be the matrix whose rows consist of the evaluations of the terms in $\mathsf{N}(\mathbf{X}')$ in the points of $\mathbf{X}'$.
   This is a non-degenerate matrix, so it has an inverse matrix $\mathcal{H}^{-1} = (h_{r,s})$.

[2] Note that if we have h_j possible values for the j-th coordinate, for each $1 \leq j \leq n$, $h = |\mathbf{X}'| = h_1 \cdots h_n \geq S$. We suppose to append to $\mathbf{X}$ the points of $\mathbf{X}' \setminus \mathbf{X}$.

4. Associate to the s-th column of $\mathcal{H}^{-1}$ the polynomial

$$p_s = \sum_{r=1}^{S} h_{r,s} x^{\alpha_r}.$$

5. $J := (p_{S+1}, ..., p_h, \gamma_1, ..., \gamma_n) = I$ even if in general this system of generators is *not* a Groebner basis for I.

Example 2.2.3. Consider the polynomial ring $\mathbf{k}[x, y]$, equipped with the lexicographical order induced by $1 < x < y$, take the simple set $\mathbf{X_0} = \{(0,0), (1,0), (1,1)\}$ and complete it to $\mathbf{X_0'} = \{(0,0), (1,0), (1,1), (0,1)\}$. Since there are 2 possible values for each coordinate, it is clear that $|\mathbf{X_0'}| = 4 > 3 = |\mathbf{X_0}|$.
We can compute (using Singular) $I' = (x^2 - x, y^2 - y)$, $\mathsf{N}_< = \{1, x, y, xy\}$. The first matrix is

$$\mathcal{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

and it is not degenerate, $det(\mathcal{H}) = -1$; the inverse matrix is

$$\mathcal{H}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & -1 \end{pmatrix}.$$

So, adding to the generators of I' the polynomial $p_4 = y - xy$ we obtain a system of generators for I[3].

If we know a Groebner basis $\mathcal{G}$ and the Groebner escalier N of I, the solution of problem 1 is trivial:

1. consider $\mathcal{H}$ and its inverse $\mathcal{H}^{-1}$;

2. let $\chi =^t (\chi_1, ..., \chi_S)$;

3. the required polynomial is $p = \sum_{j=1}^{m} (\mathcal{H}^{-1}\chi)_j x^{\alpha_j}$.

[3] In this oversimplified situation it holds that $\{x^2 - x, y^2 - y, y - xy\}$ is actually a Groebner basis of I. But this is not true in general.

Example 2.2.4. Consider the same $\mathbf{X_0}$ of example 2.2.3 and $\chi = (1, 2, 3)$. In this situation the reduced Groebner basis $\{x^2 - x, y^2 - y, xy - y\}$ is very simple and the Groebner escalier is likewise simple: $N = \{1, x, y\}$. We have then

$$\mathcal{B} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix},$$

and $det(\mathcal{B}) = -1$, while the inverse matrix is

$$\mathcal{B}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}.$$

The required polynomial is then $p = x + y + 1$.

In order to obtain the same result we can also proceed in another way:

1. attach χ_i as $n+1$-th coordinate of P_i, for each $i = 1, ..., S$, forming a new set $\mathbf{Y}$;
2. add a new variable t to the ring supposing it much bigger w.r.t. the other ones;
3. compute the reduced Groebner basis of $I(\mathbf{Y})$ and take the polynomial q whose leading term is t: $(-1) \cdot (q - t)$ is our required p.

Example 2.2.5. Referring to examples 2.2.3, 2.2.4, we take again $\mathbf{X_0}$ and construct $\mathbf{Y} = \{(0, 0, 1), (1, 0, 2), (1, 1, 3)\}$ from it.
We also take $\chi = (1, 2, 3)$. We have $I(\mathbf{Y}) = (x^2 - x, xy - y, y^2 - y, t - y - x - 1)$, so, as we expected, $p = x + y + 1$.

2.2.2 Matrices and problem (1).

Until now, for their purposes, Cerlienco-Mureddu required the knowledge of the Groebner escalier that we have always computed by using the reduced Groebner basis of the treated ideal.
In section 3.3 of [20], Cerlienco-Mureddu state a one-to-one correspondence between the bases of the quotient algebra and the nonzero order S minors of the matrix $\mathcal{H}'$ obtained as $\mathcal{H}$, but only using the points in $\mathbf{X}$.
If A is one of such minors, $B_A := \{b_1, ..., b_S\}$ is the set of terms corresponding to A's columns and we take B_A as a basis of the quotient algebra.

We then find $y = p(x_1, ..., x_n)$ using

$$\begin{pmatrix} b_1 \dots b_S & y \\ & \alpha_1 \\ A & \vdots \\ & \alpha_S \end{pmatrix} = 0$$

If the chosen minor is somehow "convenient"[4] one can use it to compute the reduced Groebner basis of our ideal of points in the following way.

1. Take $\mathsf{G} = \{\tau_1, ..., \tau_l\}$ the monomial basis for the semigroup ideal $\mathcal{T} \setminus \mathsf{N}(I(\mathbf{X}))$.
2. Denote $C^{(i)} := \{b_1, ..., b_S, \tau_i\}$ and by D^i the matrix whose first row is $C^{(i)}_<$, while the other ones are the rows of $C^{(i)}_<(\mathbf{X})$[5]. Let then $g_i = det(D^i)$.
3. The reduced Groebner basis is $\{g_1, ..., g_l\}$.

Example 2.2.6. Take again the set $\mathbf{X_0}$. $G = \{x^2, xy, y^2\}$. We then have to define three matrices:

$$D^1 = \begin{pmatrix} 1 & x & y & x^2 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

whose determinant is $g_1 = -x^2 + x$,

$$D^2 = \begin{pmatrix} 1 & x & y & xy \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

whose determinant is $g_2 = -xy + y$ and

$$D^3 = \begin{pmatrix} 1 & x & y & y^2 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

whose determinant is $g_3 = -y^2 + y$. We have obtained the reduced Groebner basis of the ideal $I(\mathbf{X})$.

[4]Convenient means that if we take another set of S terms (among the ones in the Groebner escalier of the elementary ideal $I(\mathbf{X}')$) which are smaller or equal to the maximal in B_A, the determinant of the corresponding minor (evaluating in $\mathbf{X}$) is 0.

[5]in Cerlienco-Mureddu notation, it means evaluating the terms of C in $\mathbf{X}$. The j-th row of C is the evaluation of te terms in C in $P_j \in \mathbf{X}$

2.2.3 The combinatorial algorithm.

Cerlienco-Mureddu define a purely combinatorial algorithm in order to produce directly the lexicographical Groebner escalier from $\mathbf{X}$.
More precisely, they prove that there is a one-to-one correspondence which sends each point of $\mathbf{X}$ to a term in the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$. The idea underlying the algorithm is the following: take a point $P_i \in \mathbf{X}$, $i > 1$[6] and find the exponent d_{is} of the maximal variable x_s, $s \leq n$ appearing in the term to be associated to P_i by Φ. It consists finding the maximal length s' for a sequence of coordinates- from the first on- shared by P_i with a previous point:$s = s' + 1$. Then, among the points sharing the first s' coordinate with P_i, we choose the one with maximal index (say P_m: in the algorithm we only keep trace of the index m). It is $d_{is} = d_{ms} + 1$. This means that our P_i will be drawn in the first range w.r.t $x_{s+1}, ..., x_n$ and the exponent of x_s gives us also the x_s-range in which to put it.
Then the algorithm restricts to this range and proceeds in the same way with $x_1, ..., x_{s-1}$[7].
Repeating all the procedure we are able, in a finite number of steps, to settle P_i and obtain $\Phi(P_i)$ via the list of exponents of the corresponding term.
More precisely we have algorithm 4.

Example 2.2.7. Take the set, proposed for the first time by Gao-Rodrigues-Stroomer in [39]
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$
$(3,1,1,2), (3,1,2,2), (3,1,2,3), (3,3,1,1), (3,4,1,1), (3,4,1,2)\}$,
and consider the ring $\mathbf{k}[x,y,z,t]$, equipped with the lexicographical order induced by $1 < x < y < z < t$.
Consider the points numbered in the order in which they appear in $\mathbf{X_1}$.
We run now Cerlienco-Mureddu algorithm on $\mathbf{X_1}$:
$P_1 = (1,1,2,3)$: it is the first point so it corresponds automatically to 1.
$P_2 = (1,1,2,4)$: $s = 4$, $m = 1$, so $\Phi(P_2) = x^?y^?z^?t$; then we repeat the algorithm on $\mathcal{Q} = \{(1,1,2)\}$ obtaining $\{1\}$ and then $\Phi(P_2) = t$.
$P_3 = (1,1,2,5)$: $s = 4$, $m = 2$, so $\Phi(P_3) = x^?y^?z^?t^2$; we then repeat the algorithm on $\mathcal{Q} = \{(1,1,2)\}$ obtaining $\{1\}$ and then $\Phi(P_3) = t^2$.
$P_4 = (1,2,1,1)$: $s = 2$, $m = 3$, so $\Phi(P_4) = x^?yz^0t^0 = x^?y$; we then repeat the algorithm on $\mathcal{Q} = \{(1)\}$ obtaining $\{1\}$ and then $\Phi(P_4) = y$.
$P_5 = (1,2,1,2)$: $s = 4$, $m = 4$, so $\Phi(P_5) = x^?y^?z^?t$; we then repeat the algorithm on $\mathcal{Q} = \{(1,1,2), (1,2,1)\}$.

[6] Obviously, $\Phi(P_1) = 1$, since $I(\{P_1\})$ is maximal and so its Groebner escalier is clearly the singleton $\{1\}$. We take the first point as a base case for this inductive algorithm.
[7] So we project the points of the restricted range with π_{s-1}

Algorithm 4 Cerlienco-Mureddu algorithm.

procedure CEMU($\underline{\mathbf{X}}$) $\to \Phi(\underline{\mathbf{X}})$
  if $S = 1$ **then**
    $\Phi(\underline{\mathbf{X}}) := \{\mathbf{d_1}\} = \{(0, ..., 0)\}$.
  end if
  if $1 < S$ **then**
    $\mathbf{d}_1 = (0, ..., 0)$ ▷ This is the base step for the algorithm
    for $l = 2$ to S **do**
      $s = \sigma(P_l, \underline{\mathbf{X}})$.
      for $i = n$ to 1 **do**
        if $i > s$ **then**
          $d_{li} = 0$.
        end if
        if $i = s$ **then**
          find the maximal integer m, $(1 \leq m \leq l-1)$ s.t $\pi_{s-1}(P_m) = \pi_{s-1}(P_l)$, $\pi^{s+1}(\mathbf{d}_m) = (0, ..., 0) = \pi^{s+1}(\mathbf{d}_l)$. ▷ P_m is the σ *-antecedent* of P_l w.r.t. $(P_1, \ldots, P_{l-1})$, $\Phi((P_1, \ldots, P_{l-1}))$.
          $d_{ls} = d_{ms} + 1$.
        end if
        if $i < s$ **then**
          $\mathcal{W}(P_l, \underline{\mathbf{X}}) := \{P \in \underline{\mathbf{X}} | \, \Phi(P) = \mathbf{d} = (*, ...*, d_{ls}, 0, ..., 0), \} = \{P_{j1}, ..., P_{jr}\}$.
          $\mathcal{Q} := \pi_{s-1}(\mathcal{W}(P_l, \underline{\mathbf{X}}))$. ▷ If $h < r = |\mathcal{W}(P_l, \underline{\mathbf{X}})|$, then $\pi_{s-1}(P_{jh}) \neq \pi_{s-1}(P_l)$. Moreover, since Φ is inductive, if $h < k \leq r$ then $\pi_{s-1}(P_{jh}) \neq \pi_{s-1}(P_{jk})$. $|\mathcal{Q}| = |\mathcal{W}(P_l, \underline{\mathbf{X}})| = r < l$.
          $\Phi(\mathcal{Q}) = \text{CEMU}(\mathcal{Q}) := \{\widetilde{\mathbf{d}_1}, .., \widetilde{\mathbf{d}_r}\}$
          $\pi_{s-1}(\mathbf{d}_l) = \widetilde{\mathbf{d}_r}$. ▷ We know $\Phi(\underline{\mathcal{Q}}) = (\widetilde{\mathbf{d}_1}, .., \widetilde{\mathbf{d}_r})$ and $\forall 1 \leq i < r, \widetilde{\mathbf{d}_i} = \pi_{s-1}(\mathbf{d}_{ji})$.
          break.
        end if
      end for
    end for
  end if
return $\Phi(\underline{\mathbf{X}})$.
end procedure

$P_{5,1} = (1,1,2)$: it is the first point, corresponding to 1.

$P_{5,2} = (1,2,1)$: $s = 2$, $m = 1$, so $\Phi(P_{5,2}) = x^{?}y^{1}z^{0}$; we then repeat the algorithm on $\mathcal{Q}' = \{1\}$, obtaining $\{1\}$ and then $\Phi(P_{5,2}) = y$.

We obtain by recursion the partial result $\{1, y\}$ and then $\Phi(P_5) = yt$.

$P_6 = (1,2,2,1)$: $s = 3$, $m = 5$, so $\Phi(P_6) = x^{?}y^{?}z^{1}t^{0} = x^{?}y^{?}z$; we then repeat the algorithm on $\mathcal{Q} = \{(1,2)\}$, obtaining $\{1\}$ and then $\Phi(P_6) = z$.

$P_7 = (1,2,2,2)$: $s = 4$, $m = 6$, so $\Phi(P_7) = x^{?}y^{?}z^{?}t$; we then repeat the algorithm on $\mathcal{Q} = \{(1,1,2),(1,2,1),(1,2,2)\}$.

$P_{7,1} = (1,1,2)$: it is the first point, corresponding to 1.

$P_{7,2} = (1,2,1)$: $s = 2$, $m = 1$, so $\Phi(P_{7,2}) = x^{?}y^{1}z^{0}$; we then repeat the algorithm on $\mathcal{Q}' = \{1\}$, obtaining $\{1\}$ and then $\Phi(P_{7,2}) = y$.

$P_{7,3} = (1,2,2)$: $s = 3$, $m = 2$, so $\Phi(P_{7,3}) = x^{?}y^{?}z^{1}$; we then repeat the algorithm on $\mathcal{Q}'' = \{(1,2)\}$, obtaining $\{1\}$ and then $\Phi(P_{7,3}) = z$.

We obtain by recursion the partial result $\{1, y, z\}$ and then $\Phi(P_7) = zt$.

$P_8 = (3,1,1,2)$: $s = 1$, $m = 7$, so $\Phi(P_8) = x^{1}y^{0}z^{0}t^{0} = x$.

$P_9 = (3,1,2,2)$: $s = 3$, $m = 8$, so $\Phi(P_9) = x^{?}y^{?}z^{1}t^{0} = x^{?}y^{?}z$; we then repeat the algorithm on $\mathcal{Q} = \{(1,2),(3,1)\}$:

$P_{9,1} = (1,2)$: it is the first point, so we associate 1 to it; $P_{9,2} = (3,1)$: $s = 1$, $m = 1$, so $\Phi(P_{9,2}) = xy^{0}z^{0}t^{0} = x$.

We obtain by recursion the partial result $\{1, x\}$, so $\Phi(P_9) = xz$.

$P_{10} = (3,1,2,3)$:$s = 4$, $m = 9$, so $\Phi(P_{10}) = x^{?}y^{?}z^{?}t$; we then repeat the algorithm on $\mathcal{Q}' = \{(1,1,2),(1,2,1),(1,2,2),(3,1,2)\}$.

$P_{10,1} = (1,1,2)$: it is the first point, corresponding to 1.

$P_{10,2} = (1,2,1)$: $s = 2$, $m = 1$, so $\Phi(P_{10,2}) = x^{?}y^{1}z^{0}$; we then repeat the algorithm on $\mathcal{Q}' = \{1\}$, obtaining $\{1\}$ and then $\Phi(P_{10,2}) = y$.

$P_{10,3} = (1,2,2)$: $s = 3$, $m = 2$, so $\Phi(P_{10,3}) = x^{?}y^{?}z^{1}$; we then repeat the algorithm on $\mathcal{Q}'' = \{(1,2)\}$, obtaining $\{1\}$ and then $\Phi(P_{10,3}) = z$.

$P_{10,4} = (3,1,2)$: $s = 1$, $m = 3$, so $\Phi(P_{10,4}) = x^{1}y^{0}z^{0} = x$.

We obtain by recursion the partial result $\{1, y, z, x\}$, so $\Phi(P_{10}) = xt$.

$P_{11} = (3,3,1,1)$: $s = 2$, $m = 10$, so $\Phi(P_{11}) = x^{?}y^{1}z^{0}t^{0} = x^{?}y$; we then repeat the algorithm on $\mathcal{Q} = \{1,3\}$.

$P_{11,1} = 1$: the first point is associated to 1;

$P_{11,2} = 3$: $s = 1$, then $\Phi(P_{11,2}) = x$.

We obtain by recursion the partial result $\{1, x\}$, so $\Phi(P_{11}) = xy$.

$P_{12} = (3,4,1,1)$: $s = 2$, $m = 11$, so $\Phi(P_{12}) = x^{?}y^{2}z^{0}t^{0} = x^{?}y^{2}$; we then repeat the algorithm on $\mathcal{Q} = \{3\}$, obtaining $\{1\}$, so $\Phi(P_{12}) = y^2$.

$P_{13} = (3,4,1,2)$:$s = 4$, $m = 12$, so $\Phi(P_{13}) = x^? y^? z^? t$; we then repeat the algorithm on $\mathcal{Q} = \{(1,1,2),(1,2,1),(1,2,2),(3,1,2),(3,4,1)\}$.

$P_{13,1} = (1,1,2)$: it is the first point, corresponding to 1.

$P_{13,2} = (1,2,1)$: $s = 2$, $m = 1$, so $\Phi(P_{13,2}) = x^? y^1 z^0$; we then repeat the algorithm on $\mathcal{Q}' = \{1\}$, obtaining $\{1\}$ and then $\Phi(P_{13,2}) = y$.

$P_{13,3} = (1,2,2)$: $s = 3$, $m = 2$, so $\Phi(P_{13,3}) = x^? y^? z^1$; we then repeat the algorithm on $\mathcal{Q}'' = \{(1,2)\}$, obtaining $\{1\}$ and then $\Phi(P_{13,3}) = z$.

$P_{13,4} = (3,1,2)$: $s = 1$, $m = 3$, so $\Phi(P_{13,4}) = x^1 y^0 z^0 = x$.

$P_{13,5} = (3,4,1) : s = 2$, $m = 4$, so $\Phi(P_{13,5}) = x^? y^1 z^0 = x^? y$; we repeat the algorithm on $\mathcal{Q}'' = \{1,3\}$:

$P_{13,5,1} = 1$: the first point corresponds to 1;

$P_{13,5,2} = 3$: $s = 1$, $m = 1$, so $\Phi(P_{13,5,2}) = x$.

We obtain by recursion the partial result $\{1, y, z, x, xy\}$, so $\Phi(P_{13}) = xyz$.

In conclusion, the final result is $\mathsf{N} = \Phi(\mathbf{X_1})$:

$$\begin{aligned}
\mathsf{N}[1] &= 1\\
\mathsf{N}[2] &= t\\
\mathsf{N}[3] &= t^2\\
\mathsf{N}[4] &= y\\
\mathsf{N}[5] &= yt\\
\mathsf{N}[6] &= z\\
\mathsf{N}[7] &= zt\\
\mathsf{N}[8] &= x\\
\mathsf{N}[9] &= xz\\
\mathsf{N}[10] &= xt\\
\mathsf{N}[11] &= xy\\
\mathsf{N}[12] &= y^2\\
\mathsf{N}[13] &= xyt
\end{aligned}$$

Remark 2.2.8 ([70, 79])**.** In the case of the polynomial ring in two variables, we can find in a simple way a possible Cerlienco-Mureddu-like correspondence between points and terms. Given a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\} \subset \mathbf{k}^2$, with $P_i = (a_{i1}, a_{i2})$, we compute the projection w.r.t. the first coordinate, namely $\pi_1(\mathbf{X}) = \{a_0, ..., a_{r-1}\}$ and we denote $d(i) := |\{(x_1, x_2) \in \mathbf{X},\ x_1 = a_i\}|$.

We can assume $d(1) \geq .... \geq d(r)$, up to a renumbering of the elements a_i, $i = 0, ..., r-1$. There exist values $b_{i,l}$, $i \in \{0, ..., r-1\}$, $l \in \{0, ..., d(i)-1\}$ such that

$$\mathbf{X} = \{(a_i, b_{il}),\ 0 \leq i \leq r-1,\ 0 \leq l < d(i)\}.$$

Therefore

1. $\mathsf{N}(I(\mathbf{X})) = \{x_1^i x_2^l,\ 0 \leq i \leq r-1,\ 0 \leq l < d(i)\}$;
2. $\Phi(a_i, b_{il}) = x_1^i x_2^l$.

This means *reordering the towers by height* in order to compute the tower structure.

The most important feature of Cerlienco-Mureddu algorithm is its iterativity on $\mathbf{X}$. Cerlienco-Mureddu do not study the computational complexity of their algorithm, but Lundqvist ([67]) does it, stating the following

Proposition 2.2.9 ([67])**.** The combinatorial algorithm described has complexity $O(n^2 S^2)$.

In [20] and [21], Cerlienco-Mureddu generalize their procedure to multiple points.

2.2.4 Application to the reduced Groebner basis.

In their papers [20, 21, 22], Cerlienco-Mureddu refer to the properties of of Ferrers diagrams. For a Ferrers diagram, they also employ the notion of dihedral elements in the proof of the correctness for their combinatorial algorithm.

Definition 2.2.10. If $\mathfrak{F}$ is a Ferrers diagram, an element $\mathbf{j} \in \mathbb{N}^n$ is *external dihedral* for $\mathfrak{F}$ if:

1. $\{\mathbf{i} \in \mathbb{N}^n / \mathbf{i} < \mathbf{j}\} \subseteq \mathfrak{F}$;
2. $\{\mathbf{i} \in \mathbb{N}^n / \mathbf{j} \leq \mathbf{i}\} \cap \mathfrak{F} = \emptyset$.

Definition 2.2.11. With the same notation of definition 2.2.10, an element $\mathbf{j} \in \mathbb{N}^n$ is called *internal dihedral* for $\mathfrak{F}$ if:

1. $\{\mathbf{i} \in \mathbb{N}^n / \mathbf{i} \leq \mathbf{j}\} \subseteq \mathfrak{F}$;
2. $\{\mathbf{i} \in \mathbb{N}^n / \mathbf{j} > \mathbf{i}\} \cap \mathfrak{F} = \emptyset$.

They develop an algorithm which computes the reduced Groebner basis of $I(\mathbf{X})$, inductively on $|\mathbf{X}|$.
Denote by $\mathfrak{F}$ the Ferrers diagram associated to $\mathsf{N}(I(\mathbf{X}))$[8], and let $f_1 < ... < f_s$ their external dihedral elements. The reduced Groebner basis of $I(\mathbf{X})$ has the form

$$\mathcal{G}(I(\mathbf{X})) = \{x^{f_1} - p_1, ..., x^{f_s} - p_s\},$$

[8]It is simply the set of the n-tuples of exponents corresponding to the elements of the Groebner escalier.

where p_i only contain terms smaller than x^{f_i}.
Take $P = (a_1, ..., a_n) \notin \mathbf{X}$, $\mathbf{X}' = \mathbf{X} \cup \{P\}$ and let $\mathfrak{F}', \mathcal{G}'$ the analogous sets as $\mathfrak{F}, \mathcal{G}$.
Let j the minimal index such that P is not a zero of $x^{f_j} - p_j$, then one can easily see that $\mathsf{N}(I(\mathbf{X})) \cup \{x^{f_j}\}$ is a basis for $\mathbf{k}[x_1, ..., x_n]/I(\mathbf{X}')$.
Notice that the external dihedral elements of $\mathfrak{F}$, different from f_j are external dihedral also of $\mathfrak{F}'$; the possible remaining elements of $\mathfrak{F}'$ are of the shape $f_j + e_h$ ($e_1 = (1, 0, ..., 0), ..., e_n = (0, 0, ..., 1)$). In order to find the basis $\mathcal{G}'$ we have to consider the following polynomials:

1. for each external dihedral f_i different from f_j ($i \neq j$) we have
$$g_i = x^{f_i} - p_i - \frac{A_i}{A_j}(x^{f_j} - p_j),$$
where $A_i = ev_P(x^{f_i} - p_i)$, $A_j = ev_P(x^{f_j} - p_j)$;
2. for each $f_j + e_h$, $g_{j,h} = (x_h - a_h)(x^{f_j} - p_j)$.

Actually, here they are only rewriting Moeller algorithm in the version iterative on functionals ([73], algorithm 2).
In [20], Cerlienco-Mureddu discuss how to simiplify the algorithm in the bidimensional case.

Proposition 2.2.12. Let $\mathbf{X} \subset \mathbf{k}^2$. If the points of $\mathbf{X}$ have r different x-coordinates $\rho_1, ..., \rho_r$ and there are h_i points having ρ_i as first coordinate. Assuming $h_1 \geq ... \geq h_r$, the associated order ideal is:
$$1, y, ..., y^{h_1 - 1}$$
$$x, xy, ..., xy^{h_2 - 1}$$
$$...$$
$$x^{r-1}, ..., x^{r-1}y^{h_r - 1}.$$

If we think again about the tower structure introduced above, we can interpret the proposition 2.2.12 as follows:
ordering the towers in non-increasing order by height, we obtain the Groebner escalier, under the identification defined above. See 2.2.8

Remark 2.2.13. Come now back to examples 1.4.1, 1.4.2. We can see that the towers are not non-increasingly ordered, but that, if we do it, we obtain the Groebner escalier.
If we look at example 1.4.3, we see that it is exactly the output of Cerlienco-Mureddu algorithm on
$$\mathbf{X}'' = \{(1, 0, 0), (2, 0, 0), (1, 1, 0), (2, 1, 0), (1, 0, 1), (2, 0, 1), (1, 1, 1), (2, 1, 1)\},$$

where the points are taken in the order they are listed.

In [21, 22] Cerlienco-Mureddu also state an application of the algorithm to n-linearly recursive functions.

2.3 Gao-Rodrigues-Stroomer method.

In [39], Gao-Rodrigues-Stroomer , in the special case **k** perfect field, study the relationship between the fibers $\pi_{n-1}(\mathbf{X}) \subseteq k^{n-1}$ of a given set of distinct points $\mathbf{X} \subseteq \mathbf{k}^n$ and a minimal Groebner basis for $I(\mathbf{X})$ under an elimination order for x_n.
Moreover they explain how to use their results in order to simplify systems of equations.
They "do not describe how to calculate a Groebner basis for a given set of points" (p.3), but there is a paper by Farr and Gao doing it [35], as well as, clearly, Moeller algorithm does [12, 73].
In the case where the elimination ordering is exactly the lexicographical one ($x_1 < ... < x_n$), Gao-Rodrigues-Stroomer introduced a combinatorial non-iterative algorithm in order to compute directly the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$, i.e. an alternative algorithm to the one by Cerlienco-Mureddu .
Actually they compute the Ferrers diagram $\mathfrak{F}(\mathbf{X})$ containing the exponents' lists of the terms belonging to $\mathsf{N}(I(\mathbf{X}))$.
They first make some preprocessing on the given points, namely they construct a tree associated to them and this is the step excluding iterativity.
Then, usign a "merging" procedure, they read the tree and return the Groebner escalier.
Let us examine the procedure more in details.
The first step consists to associate to $\mathbf{X}$ a tree $T(\mathbf{X})$ of height n, whose nodes are labeled with the coordinates of the points (except that the root, i.e. the 0 level node, which is simply labeled with "root").
From the root arise as many edges as the first coordinate values, from each 1 level node arise as many edges as the second coordinate values corresponding to the given first coordinate value and so on. The S leaves (one for each point) are so ordinately labeled with the n-th coordinates.
If two points share the first k coordinates, then their corresponding paths coincide from level 0 to level $k+1$.
After giving the tree construction, they define the merging procedure of Ferrers diagrams.

Procedure 2.3.1. Let $\mathfrak{F}_1, ..., \mathfrak{F}_k \subseteq \mathbb{N}^{n-1}$ be Ferrers diagrams.
For each $P = (p_2, ..., p_N) \in \mathbb{N}^{n-1}$ let $\delta(P)$ be the number of Ferrers diagrams containing P.

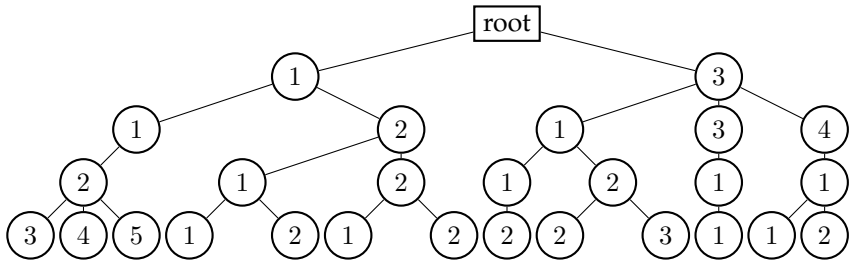
Merging these Ferrers diagrams means construct the Ferrers diagram

$$\mathfrak{F} := M(\mathfrak{F}_1, ..., \mathfrak{F}_k) = \{(j, p_2, ..., p_N) \,|\, 0 \leq j < \delta(p_2, ..., p_N)\} \subseteq \mathbb{N}^n$$

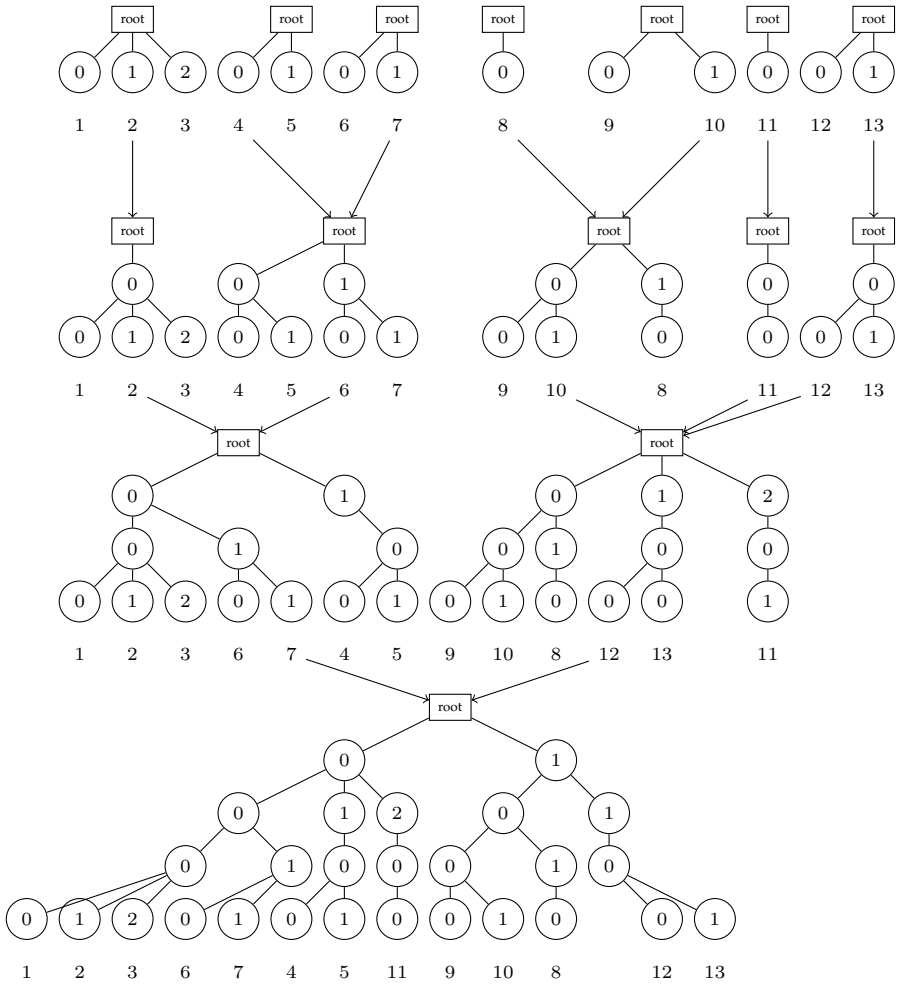
Gao-Rodrigues-Stroomer algorithm then, consists of the following three steps:

- construct $T(\mathbf{X})$;
- if $n = 1$, then $\mathfrak{F}(\mathbf{X}) = \{0, 1, ..., |\mathbf{X}| - 1\}$;
- otherwise:
  - consider the subtrees $T_1, ..., T_l$ of $T(\mathbf{X})$, obtained removing the root from it and taking the elements of the resulting subforest;
  - assume to have computed recursively $\mathfrak{F}_1, ..., \mathfrak{F}_l$, i.e. the Ferrers diagrams associated to the points drawn in $T_1, ..., T_l$;
  - $\mathfrak{F}(\mathbf{X})$ is obtained by merging $\mathfrak{F}_1, ..., \mathfrak{F}_l$.

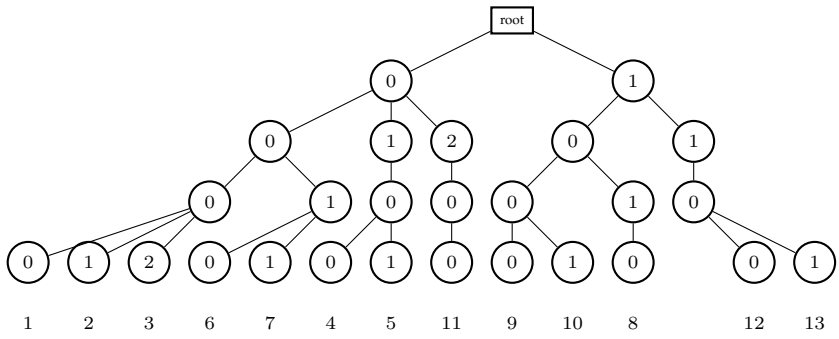
Example 2.3.2. Take (as in example 2.2.7) the set
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$
$(3,1,1,2), (3,1,2,2), (3,1,2,3), (3,3,1,1), (3,4,1,1), (3,4,1,2)\}$,
and consider the ring $\mathbf{k}[x, y, z, t]$, equipped with the lexicographical order induced by $1 < x < y < z < t$. The tree associated to the set is



The merging process works as follows. In the picture below, we represent each step of the algorithm, using arrows in order to point out what sets are merged together and what is the final result of each merging operation:



The final result is then



In both the previous pictures, the numbers not surrounded by the circles are not to be in-

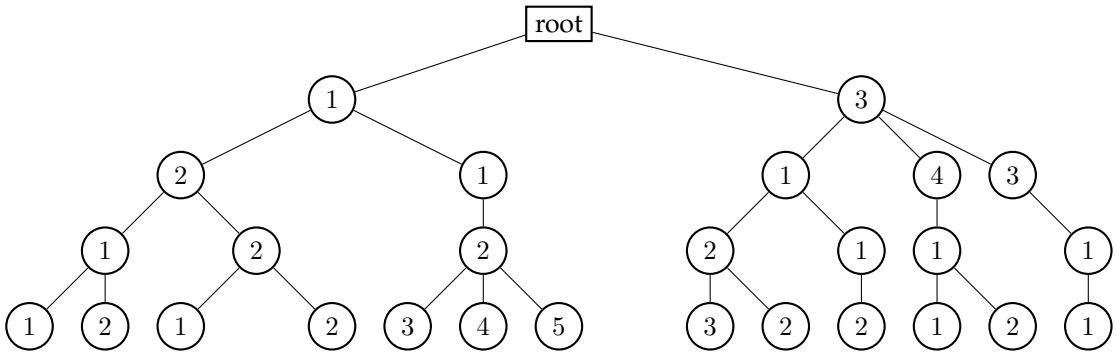
tended as nodes for some graph. They denote the indices of the points corresponding to the element of the Ferrers diagram at each step (see 2.3.3 below for more details). We summarize here the steps outlined in the picture. Start with the leaves:
$\mathfrak{F}_1 = \{0, 1, 2\}$
$\mathfrak{F}_2 = \{0, 1\}$
$\mathfrak{F}_3 = \{0, 1\}$
$\mathfrak{F}_4 = \{0\}$
$\mathfrak{F}_5 = \{0, 1\}$
$\mathfrak{F}_6 = \{0\}$
$\mathfrak{F}_7 = \{0, 1\}$
and perform the first merging step.
$\mathfrak{F}_8 = M(\mathfrak{F}_1) = \{(0,0), (0,1), (0,2)\}$
$\mathfrak{F}_9 = M(\mathfrak{F}_2, \mathfrak{F}_3) = \{(0,0), (0,1), (1,0), (1,1)\}$
$\mathfrak{F}_{10} = M(\mathfrak{F}_4, \mathfrak{F}_5) = \{(0,0), (0,1), (1,0)\}$
$\mathfrak{F}_{11} = M(\mathfrak{F}_6) = \{(0,0)\}$
$\mathfrak{F}_{12} = M(\mathfrak{F}_7) = \{(0,0), (0,1)\}$.
Now we merge again:
$\mathfrak{F}_{13} = M(\mathfrak{F}_8, \mathfrak{F}_9) = \{(0,0,0), (0,0,1), (0,0,2), (0,1,0), (0,1,1), (1,0,0), (1,0,1)\}$
$\mathfrak{F}_{14} = M(\mathfrak{F}_{10}, \mathfrak{F}_{11}, \mathfrak{F}_{12}) = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,0,1), (2,0,0)\}$
and, in conclusion,
$\mathfrak{F}(\mathbf{X}) = \mathfrak{F}_{15} = M(\mathfrak{F}_{13}, \mathfrak{F}_{14}) = \{(0,0,0,0), (0,0,0,1), (0,0,0,2), (0,0,1,0),$
$(0,0,1,1), (0,1,0,0), (0,1,0,1), (0,2,0,0), (1,0,0,0), (1,0,0,1), (1,0,1,0),$
$(1,1,0,0), (1,1,0,1)\}$, so the final result, as expected, is
$\mathsf{N}(I(\mathbf{X})) = \{1, t, t^2, z, , zt, y, yt, y^2, x, xt, xz, xy, xyt\}$.

Remark 2.3.3. Reading [39], we can notice that there is *no explicit intent* to stress a biunivocal correspondence between the points and the terms belonging to $\mathsf{N}(I(\mathbf{X}))$.
There is only one example (i.e. exactly example 2.3.2) which can be interpreted in this direction (as I did in the picture).
Moreover there is *no explicit intent* to give the output arranged in some order.
Anyway, we can notice a rather strange fact (again from example 2.3.2): the terms are ordered w.r.t. lex, but induced by $x_n < ... < x_1$ ($t < z < y < x$), while the Groebner escalier is computed using the reversed ordering $x_1 < ... < x_n$ ($x < y < z < t$).

The authors do not give any complexity analysis of their algorithm.

Remark 2.3.4. I underline here a strange fact about Gao-Rodrigues-Stroomer method.
In [35], the authors explicitly say *for the first time* their way to sort the points of the given $\mathbf{X}$,

referring to [39], and so making one think that *this is the sorting criterion also for* [39].
Actually, in [39] there is no declaration on how to decide what is the order of the i-th coordinates to be drawn at level i.
To be more precise, in [35], they say:
The details of this ordering, motivated by [39], are quite simple. If $x_1 < x_2 < ... < x_n$, then group the points first according to the x_1-coordinate; these groups are ordered in a nonincreasing order by size. Within each of the groups, repeat the process, but according to the x_2-coordinate. Continue for $x_3, ..., x_m$.
The surprising fact is that *this criterion is not followed in the only example displayed in [39]!*
Look at example 2.3.2. Level one is well arranged according to Farr-Gao's criterion, but we cannot assert the same for level 2. In fact, the subtree containing $(1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2)$ should have been drawn on the left w.r.t. the one containing $(1,1,2,3), (1,1,2,4), (1,1,2,5)$, but it is only one example of this curious fact. The tree, according to [35], should have been:



Lundqvist, Felszeghy-B. Ráth-Rónyai never say anything about it (even if their tree mirrors it), while Lederer does not display any example of the Groebner escalier's construction.

2.4 Lederer's variation.

Lederer, in [63], gives an alternative to Buchberger-Moeller algorithm, in order to compute a lexicographical Groebner basis of a zerodimensional radical ideal, basing his computation on Lagrange interpolation.
In the same paper he discusses a non-iterative method in order to compute directly the Groebner escalier.
It turns out that this method is equivalent to the one by Gao-Rodrigues-Stroomer discussed

above (for less than a reordering on the set to be "merged").
Let $\mathfrak{D}_n$ the set containing all the Ferrers diagram in $\mathbb{N}^n$ and take two elements $D, D' \in \mathfrak{D}_n$. Lederer defines their sum $D + D' \in \mathfrak{D}_n$:

$$D + D' := \{d \in \mathbb{N}_0^n \mid \hat{d} \in \hat{p}(D) \cup \hat{p}(D'),\ d_1 < |\hat{p}^{-1}(\hat{d}) \cap D| + |\hat{p}^{-1}(\hat{d}) \cap D'|\}.$$

Then he gives a representation of the summation operator: "*Draw a coordinate system of* $\mathbb{N}_0^n$ *and insert* D*. Place a translate of* D' *somewhere on the* 1*-axis. The translate has to be sufficiently far out, so that* D *and the translate of* D' *do not intersect. Then take the elements of the translate of* D' *and drop them down along the* 1*-axis until they lie on top of an element of* D*, just as in the popular game Connect*4*, which might be known to one reader or the other. The result is* $D + D'$*.*"
In conclusion the summation of two Ferrers diagrams consists of make one "slide on the other", only avoiding the overlapping of elements.
Clearly the summation (which is commutative and associative!) can be extended to more than two Ferrers diagrams.

Remark 2.4.1. It is very simple to notice that the summation defined above is totally equivalent to the merging operation, while taking away a coordinate means "restrict to a subtree", as Gao-Rodrigues-Stroomer do.

These informations are the only ones needed in order to compute the Groebner escalier. Lederer, given $\mathbf{X}$, proceeds by induction over n.

- ✓ If $n = 1$, $\mathfrak{F}(\mathbf{X}) = \{1, ..., |\mathbf{X}| - 1\}$.
- ✓ In order to pass from $n - 1$ to n, proceed as follows.
  - ⋆1. Take $\forall a_1 \in p(\mathbf{X})$ the set $H(a_1) = p^{-1}(a_1) \cap \mathbf{X}$.
  - ⋆2. Consider $H(a_1)$ as a subset of k^{n-1} via the projection map $\hat{p}$: in this way $\mathfrak{F}(H(a_1)) \subseteq \mathfrak{D}_{n-1}$ is defined by the induction hypothesis.
  - ⋆3. Identify each $\mathfrak{F}(H(a_1))$ as an element of $\mathfrak{D}_n$, adding a 0 as first component to each element of it.
  - ⋆4. Set $\mathfrak{F}(\mathbf{X}) = \sum_{a_1 \in p(\mathbf{X})} \mathfrak{F}(H(a_1))$.

Example 2.4.2. Take again $\mathbf{X_1}$, as in examples 2.2.7, 2.3.2, namely
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$
$(3,1,1,2), (3,1,2,2), (3,1,2,3), (3,3,1,1), (3,4,1,1), (3,4,1,2)\}$.
We can perform Lederer's algorithm on this set. Since we will need to compute $H(a_1)$ in more than one nested step, we will use superscripts $'$ in order to distinguish the different

steps.
The first coordinates of the points in $\mathbf{X_1}$ are $1, 3$:
$H(1) = \{P_1, ..., P_7\}$
$H(3) = \{P_8, ..., P_{13}\}$. Denote by $P(i)$ the set containing the indexes of the points in $H(i)$.
We should compute $\mathfrak{F}(H(1)) + \mathfrak{F}(H(3))$, but we need to know the addenda.
Focus on $H(1)$ (forget $H(3)$, for the moment), thinking about it in $\mathbf{k}^3$.
$H(1) = \{(1,2,3), (1,2,4), (1,2,5), (2,1,1), (2,1,2), (2,2,1), (2,2,2)\}$, corresponding to the set of indexes (of the associated points)
$P(1) = \{1, 2, 3, 4, 5, 6, 7\}$ since the first coordinate values are 1 and 2 we will need to work with
$H'(1) = \{(1,2,3), (1,2,4), (1,2,5)\}$
$P'(1) = \{1, 2, 3\}$
$H'(2) = \{(2,1,1), (2,1,2), (2,2,1), (2,2,2)\}$
$P'(2) = \{4, 5, 6, 7\}$ and $\mathfrak{F}(H(1)) = \mathfrak{F}(H'(1)) + \mathfrak{F}(H'(2))$.
Focus on $H'(1)$ thinking about it in $\mathbf{k}^2$:
$H'(1) = \{(2,3), (2,4), (2,5)\}$.
It has only 2 as first coordinate, so we have only $H''(2) = \{3, 4, 5\}$, $P''(2) = \{1, 2, 3\}$.
The Ferrers diagram $\mathfrak{F}(H''(2)) = \{0, 1, 2\}$ (corresponding to the points individuated by the elements of $P''(2)$, taken in order) can be thought in $\mathbf{k}^2$ as explained in $\star 3$., so
$\mathfrak{F}(H'(1)) = \{(0,0), (0,1), (0,2)\}$.
Now consider $H'(2)$ in $\mathbf{k}^2$:
$H'(2) = \{(1,1), (1,2), (2,1), (2,2)\}$.
Its first coordinates are $1, 2$, so $\mathfrak{F}(H'(2)) = \mathfrak{F}(H''(1)) + \mathfrak{F}(H''(2))$, with
$H''(1) = \{1, 2\}$
$P''(1) = \{4, 5\}$ $H''(2) = \{1, 2\}$ $P''(2) = \{6, 7\}$.
We have $\mathfrak{F}(H''(1)) = \{0, 1\} = \mathfrak{F}(H''(2))$ and we see it in $\mathbf{k}^2$, obtaining
$\mathfrak{F}(H'(2)) = \{(0,0), (1,0), (0,1), (1,1)\}$.
While summing we take the elements in order i.e. for example we have two couples with 0 in second place since we find 0 in both $\mathfrak{F}(H''(1))$ and $\mathfrak{F}(H''(2))$, so we associate $(0,0)$ to P_4 (4 is associated to the 0 element of the *first* Ferrers diagram) and $(1,0)$ to P_6 (6 is associated to the 0 element of the *second* Ferrers diagram).
We always behave this way for the sum.
Finally we can compute
$\mathfrak{F}(H(1)) = \{(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,0,1), (0,1,1), (0,0,2)\}$, associated to the following reordering of $P(1)$:
$\{1, 4, 6, 2, 5, 7, 3\}$.

Now we focus on $H(3)$, thinking about it in $\mathbf{k}^3$.
$H(3) = \{(1,1,2),(1,2,2),(1,2,3),(3,1,1),(4,1,1),(4,1,2)\}$.
$P(3) = \{8,9,10,11,12,13\}$ Its first coordinates are $1,3,4$, so we have
$H'(1) = \{(1,1,2),(1,2,2),(1,2,3)\}$
$P'(1) = \{8,9,10\}$ $H'(3) = \{(3,1,1)\}$
$P'(3) = \{11\}$
$H'(4) = \{(4,1,1),(4,1,2)\}$
$P'(4) = \{12,13\}$.
Consider $H'(1)$ as a subset of $\mathbf{k}^2$
$H'(1) = \{(1,2),(2,2),(2,3)\}$.
Its first coordinates are 1 and 2, so $\mathfrak{F}(H'(1)) = \mathfrak{F}(H''(1)) + \mathfrak{F}(H''(2))$, with
$H''(1) = \{2\}$
$P''(1) = \{8\}$ $H''(2) = \{2,3\}$ $P''(2) = \{9,10\}$.
$\mathfrak{F}(H''(1)) = \{0\}, \mathfrak{F}(H''(2)) = \{0,1\}$, so
$\mathfrak{F}(H'(1)) = \{(0,0),(1,0),(0,1)\}$, associated to $\{8,9,10\}$.
Now take $H'(3)$ in $\mathbf{k}^2$:
$H'(3) = \{(1,1)\}$, whose associated Ferrers diagram is (by the same reasoning made before for the case of only one first coordinate) $\mathfrak{F}(H'(3)) = \{(0,0)\}$.
Take at the end $H'(4) = \{(1,1),(1,2)\}$ whose associated Ferrers diagram is obtained passing through $H''(4) = \{1,2\}$ and $P''(4) = \{12,13\}$:
$\mathfrak{F}(H'(4)) = \{(0,0),(0,1)\}$, associated to $\{12,13\}$.
$\mathfrak{F}(H(3)) = \{(0,0,0),(1,0,0),(2,0,0),(0,1,0),(0,0,1),(1,0,1)\}$ (associated to $\{8,11,12,9,10,13\}$) and
$\mathfrak{F}(\mathbf{X}) = \mathfrak{F}(H(1)) + \mathfrak{F}(H(3)) = \{(0,0,0,0),(1,0,0,0),(0,1,0,0),(1,1,0,0),$
$(0,0,1,0),(1,0,1,0),(0,0,0,1),(0,0,0,1),(0,1,0,1),(1,1,0,1),(0,0,1,1),$
$(0,0,0,2),(0,2,0,0)\}$.
In conclusion
$\mathsf{N}(I(\mathbf{X})) = \{1,x,y,xy,z,xz,t,xt,yt,xyt,zt,t^2,y^2\}$ is associated to
$\{1,8,4,11,6,9,2,10,5,13,7,3,12\}$.

Remark 2.4.3. Reading [63], as in [39], we can notice that there is *no explicit intent* to stress a biunivocal correspondence between the points and the terms belonging to $\mathsf{N}(I(\mathbf{X}))$.
Actually it can be done (as I tried to in example 2.4.2), while defining the sets $P(i)$'s and the way to combine them w.r.t. the sum.
Moreover there is *no explicit intent* to give the output arranged in some order.

2.5 The Lex Game.

The mathematicians Felszeghy-B. Ráth-Rónyai , in [37], introduce the so called "Lex Game", which leads to a non-iterative combinatorial algorithm in order to compute the Groebner escalier of $I(\mathbf{X})$, the ideal of a finite set $\mathbf{X}$ of distinct points, w.r.t the lexicographical order, induced by $x_n < ... < x_1$.
They do *not* compute a Groebner basis of $I(\mathbf{X})$ (but they cite a couple of papers studying it, namely [49, 58]), focusing their efforts on computing the Groebner escalier of $I(\mathbf{X})$, when $\mathbf{X}$ is a set of points admitting as components only $0, 1$ and having the number of ones (Hamming weight) in a fixed $D \subseteq \mathbb{Z}$.
In the same paper is stated a formula for triangular polynomials and also another formula which permits to compute the normal form of a polynomial using the separators.
The "Lex Game", from which their reasoning starts, is a game with two players (Lea and Stan), consisting of the following rules. Take a field $\mathbf{k}$, a finite set $\emptyset \neq V \subseteq \mathbf{k}^n$ and $w = (w_1, ..., w_n) \in \mathbb{N}^n$.
V and w are "public", the players know them.
Lea's goal is to guess the element $v \in V$ which Stan is thinking about. She has w_n attempts in order to guess v_n and she wins if she manages to do it; if not, Stan reveals v_n and Lea tries with v_{n-1}.
Lea wins if guesses a v_i right, while Stan wins if he has to reveal v_1.
Stan's strategy is to keep saying "no" as long as the suffix known to Lea is consistent with some $v \in V$.
It turns out that Stan is able to win this way if and only if $x^w \in \mathsf{N}(I(V))$ and this leads to the study of the Groebner escalier.
A very precise description of the algorithm, together with a full example and a complexity study can be found in [67].
The first step consists on a preprocessing on the given points, in order to associate them a tree, called "point trie" by Lundqvist.
Let us equip $\mathbf{k}$ with an equivalence relation, denoted by $=$ and extend it to $\mathbf{k}^n$ by $a = (a_1, ..., a_n) = (b_1, ..., b_n) = b$ if $a_i = b_i$, $\forall i \in \{1, ..., n\}$.

Definition 2.5.1. The *witness* of two different n-tuples a, b is the minimal i such that $a_i \neq b_i$.

Consider now our points $P_1, ...P_S \in \mathbf{X} \subseteq \mathbf{k}^n$ and denote by Σ_i the the set of equivalence classes of $\pi_i(P_j), i = 1, ..., n,\ j = 1, ..., S$. We represent an equivalence class as a set containing the indices of the points in the class, instead of taking trace of the points. We usually order the classes by size, even if the algorithm works for any other ordering.
Clearly $\Sigma_0 = \{\{1, ...S\}\}$, $\Sigma_n = \{\{1\}, ..., \{S\}\}$, $|\Sigma_n| = S$

Definition 2.5.2. The *witness list* is the set W of all $i \in \{1, ..., n\}$ such that $\Sigma_{i-1} \neq \Sigma_i$, i.e. the set of witnesses.

Definition 2.5.3. The *witness matrix* is an upper-triangular matrix $C = (c_{ij})$ with elements in $W \cup \{0\}$ such that , for $i < j$, the value c_{ij} is the witness of v_i and v_j.

Using the Σ_i's we can represent the points in a trie structure (namely the point trie). More precisely we label the vertices with the elements of Σ_i's and there is an edge from $\Sigma_{i,k} \in \Sigma_i$ to $\Sigma_{i+1,h}, \in \Sigma_{i+1}$ when $\Sigma_{i+1,h} \subseteq \Sigma_{i,k}$. Such an edge is labeled $v_{i+1,j}$ for some $j \in \Sigma_{i+1,h}$.
This way, we have fixed a one-to-one correspondence between the elements of $\mathbf{X}$ and the paths from the root to the leaves in the tree.
We point out that the point trie is constructed *iteratively* on the points of the given set.
Once the point trie is constructed, we have to read it, constructing a new trie, the "lex trie", from which is possible to recover the Groebner escalier. We proceed in the following way.

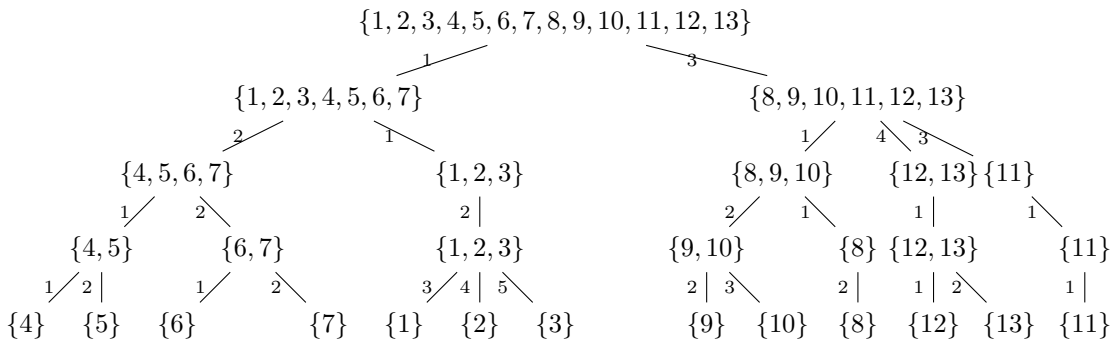
- Fix some level $h > 0$ and call $v_0, ..., v_j$ the set of vertices on level h (at level 0 we have $v_0 = \{1, ..., S\}$).
- For a class $\{i_1, ..., i_k\} \in \Sigma_{n-h}$ we let $v_{a,b} = v_{a,b} \cup \{i_k\}$ if $i_k \in v_a$ and exactly b elements in $\{i_1, ..., i_{k-1}\}$ also belong to v_a.
- The vertex set of level $h + 1$ consists of the nonempty $v_{a,b}$.
- If $v_{a,b} \neq \emptyset$, there is an edge b between v_a and $v_{a,b}$.

This new construction is *no more iterative*: we need to know all the elements in the given set and their structure summarized in the point tree in order to get the lex trie.

Remark 2.5.4. Neither Felszeghy-B. Ráth-Rónyai nor Lundqvist say to have intent to define a one to one correspondence between points of $\mathbf{X}$ and terms in $\mathsf{N}(I(\mathbf{X}))$. Anyway, this correspondence is clearly defined in their examples, namely in the lex trie construction (see 2.5.5 below).

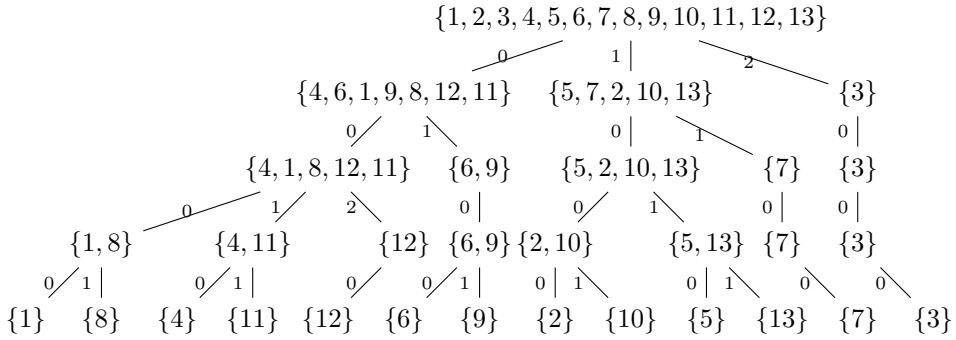
Example 2.5.5. Take the set
$\mathbf{X_2} = \{(3,2,1,1), (4,2,1,1), (5,2,1,1), (1,1,2,1), (2,1,2,1), (1,2,2,1), (2,2,2,1),$
$(2,1,1,3), (2,2,1,3), (3,2,1,3), (1,1,3,3), (1,1,4,3), (2,1,4,3)\}$,
and the polynomial ring $\mathbf{k}[x, y, z, t]$, equipped with the lexicographical order induced by $t < z < y < x$.
Working with this set $\mathbf{X_2}$ it is the same as working with
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$

$(3,1,1,2),(3,1,2,2),(3,1,2,3),(3,3,1,1),(3,4,1,1),(3,4,1,2)\}$,
i.e. reversing the ordering of the coordinates of the points.
Let us first construct the point trie, passing through the Σ_i's, containing the equivalence classes, ordered by size.
This ordering is not explicitly stated but actually Felszeghy-B. Ráth-Rónyai and Lundqvist use it in the examples.
$\Sigma_0 = \{\{1,2,3,...,13\}\}$
$\Sigma_1 = \{\{1,2,3,4,5,6,7\},\{8,9,10,11,12,13\}\}$
$\Sigma_2 = \{\{4,5,6,7\},\{1,2,3\},\{8,9,10\},\{12,13\},\{11\}\}$
$\Sigma_3 = \{\{4,5\},\{6,7\},\{1,2,3\},\{9,10\},\{8\},\{12,13\},\{11\}\}$
$\Sigma_4 = \{\{4\},\{5\},\{6\},\{7\},\{1\},\{2\},\{3\},\{9\},\{10\},\{8\},\{12\},\{13\},\{11\}\}$



Now we proceed with the lex trie construction:
$v_0 = \{1,2,...,13\}$
$h = 1$: iteration on Σ_3:
$v_{0,0} = \{4,6,1,9,8,12,11\} =: v_0$
$v_{0,1} = \{5,7,2,10,13\} =: v_1$
$v_{0,2} = \{3\} =: v_2$
$h = 2$: iteration on Σ_2:
$v_{0,0} = \{4,1,8,12,11\} =: v_0$
$v_{0,1} = \{6,9\} =: v_1$
$v_{1,0} = \{5,2,10,13\} =: v_2$
$v_{1,1} = \{7\} =: v_3$
$v_{2,0} = \{3\} =: v_4$
$h = 3$: iteration on Σ_1:
$v_{0,0} = \{1,8\} =: v_0$
$v_{0,1} = \{4,11\} =: v_1$

$v_{0,2} = \{12\} =: v_2$
$v_{1,0} = \{6, 9\} =: v_3$
$v_{2,0} = \{2, 10\} =: v_4$
$v_{2,1} = \{5, 13\} =: v_5$
$v_{3,0} = \{7\} =: v_6$
$v_{4,0} = \{3\} =: v_7$
$h = 4$: iteration on Σ_0:
$v_{0,0} = \{1\}$
$v_{0,1} = \{8\}$
$v_{1,0} = \{4\}$
$v_{1,1} = \{11\}$
$v_{2,0} = \{12\}$
$v_{3,0} = \{6\}$
$v_{3,1} = \{9\}$
$v_{4,0} = \{2\}$
$v_{4,1} = \{10\}$
$v_{5,0} = \{5\}$
$v_{5,1} = \{13\}$
$v_{6,0} = \{7\}$
$v_{7,0} = \{3\}$
The lex trie is then



Lastly, the Groebner escalier is
$\mathsf{N}(I(\mathbf{X})) = \{1, x, y, xy, y^2, z, xz, t, xt, yt, xyt, zt, t^2\}$, corresponding to the following reordering of our point set: $\{1, 8, 4, 11, 12, 6, 9, 2, 10, 5, 13, 7, 3\}$, the order of the lex trie's leaves, read from left to right.

The complexity of the Lex Game algorithm by Felszeghy-B. Ráth-Rónyai has been stud-

ied both by Lundqvist and by the authors themselves.
The last conclusive bound they found for the complexity is

$$O(nS + S\min(S, nr)),$$

where S is the number of points in the given finite set $\mathbf{X}$ and n the number of variables in the ring.
This is actually the complexity of the (iterative) construction of the point trie, since the construction of the lex trie is $O(nS)$.

CHAPTER 3

The original Axis of Evil Theorem.

3.1 Introduction

In this chapter we begin to face the problem of "*constructing a linear factorization of a lexicographical Groebner basis*" for zerodimensional radical ideals.
Initially in [2] and then in [69, 70, 71], M.G. Marinari and T. Mora studied the structure of a zerodimensional ideal I, especially in the case in which $I = \sqrt{I}$ and its Macaulay basis $\mathcal{B}(I)$ consists of the evaluations at a finite set of distinct points $\mathbf{X}$ (see also [79]).
The obtained result, named "Axis of Evil theorem" by T. Mora in some lecture notes soon after, presents a precise description of the structure of a zerodimensional ideal.
In this setting, this theorem represents, to all intents and purposes, an enhancement for the description of the Groebner basis of an ideal in $\mathbf{k}[x_1, x_2]$ given by Lazard in [62].
The theorem says that in a restricted case which includes the radical one[1], for each term $\tau := x_1^{d_1} \cdots x_n^{d_n}$ belonging to the monomial basis $\mathsf{G}(I)$ of the initial ideal of I, it is possible to produce linear factors $\gamma_{m\delta\tau} := x_i - f(x_1, \ldots, x_{i-1}), 1 \leq m \leq n, 1 \leq \delta \leq d_m$ such

[1] The most general version of the Axis of Evil Theorem holds for Cerlienco-Mureddu ideals (see 1.2.17).

that the polynomials $f_\tau := \prod_{m=1}^{n} \prod_{\delta=1}^{d_m} \gamma_{m\delta\tau}$ form a minimal lexicographical Groebner basis of I; each such factors were obtained by producing an appropriate decomposition of the given Macaulay basis $\mathcal{B}(I) = \bigsqcup_{m=1}^{n} \bigsqcup_{\delta=1}^{d_m} \mathsf{S}_{m\delta}(\tau)$ and interpolating over the monomial set obtained applying Cerlienco-Mureddu Algorithm over the set of functionals $\mathsf{S}_{m\delta}(\tau)$.
We quote here the original statement of the Axis of Evil theorem as in [2]

Theorem 3.1.1. Let $\mathbf{X} = \{P_1, ..., P_s\} \subset \mathbf{k}^n$ be a finite set of points
$I \subset \mathcal{P}$ the radical ideal whose roots are the elements in $\mathbf{X}$, $<$ the lexicographical order on $\mathcal{P}$
$\mathsf{N} := \mathsf{N}_<(I)$ the result of Cerlienco-Mureddu Correspondence
$\mathsf{G}_<(I) := \{\tau_1, ..., \tau_r\}$ the monomial basis of $T_<(I) := \mathcal{T} \setminus \mathsf{N}$, $\tau_i := x_1^{d_{1i}} \cdots x_\nu^{d_{\nu i}}$ for each i.
Then there is a combinatorial algorithm such that letting for each i, m, δ,
$\mathsf{N}_{m\delta i} := \mathsf{N}(\mathbf{X}_{m\delta i})$ be the result of Cerlienco-Mureddu Correspondence
$\gamma_{m\delta i} := x_m + \sum_{\omega \in \mathsf{N}_{m\delta i}} c(\gamma_{m\tau}, \omega)\omega$ the unique polynomial (computable by interpolation) s.t.
$\gamma_{m\delta i}(x) = 0$ for all $x \in \mathbf{X}_{m\delta i}$ and
$\gamma_{mi} = \prod_\delta \gamma_{m\delta i}$
$p_i := \gamma_{\nu i}$
$l_i := \prod_{j=1}^{\nu-1} \gamma_{ij} \in \mathbf{k}[x_1, ...x_{\nu-1}]$
$H_i := l_i p_i$
and it holds:

1. $\{H_1, ..., H_r\}$ is a (not-reduced) minimal Groebner basis of I;
2. if j_ν is the value such that $\tau_{j_\nu} < x_{\nu+1} \leq \tau_{j_\nu+1}$, then $\{H_{\tau_1}, ..., H_{\tau_{j_\nu}}\}$ is a minimal Groebner basis of $I \cap \mathbf{k}[x_1, ..., x_\nu]$;
3. if $j(\nu\delta)$ is the value such that $\tau_{j(\nu\delta)} < x_{\nu+1}^\delta \leq \tau_{j(\nu\delta)+1}$; then $\{l_1, ..., l_{j(\nu\delta)}\}$ is a Groebner basis of $J(Y_{\nu\delta})$;
4. for each i, $2 \leq i \leq r$, $p_i \in (H_j, j < i) : l_i$.

The theorem 3.1.1 above has been proved by T. Mora in [79], as a consequence of Moeller algorithm and interpreted as a sort of "interpolating variation" of Cerlienco-Mureddu algorithm. In the book [79], the theorem above is presented in its most generalized version for Cerlienco-Mureddu ideals (see definition 1.2.17).
In this thesis, we want to provide a constructive proof for the existence of the factorization in the radical case.
Such a proof turns out to be naturally associated to an algorithm (i.e. algorithm 5), allowing to get concretely the "linear" factorization of a zerodimensional radical ideal I, starting from the finite set of distinct points $\mathbf{X} = V(I)$.

We will call algorithm 5 *Axis of Evil algorithm* from now on.
In order to compute the factorization we need to calculate the Groebner escalier $\mathsf{N} = \mathsf{N}(I)$, directly from the elements in $\mathbf{X}$ and the monomial basis $\mathsf{G} = \mathsf{G}(I)$ from $\mathsf{N} = \mathsf{N}(I)$. As seen in chapter 2, the first problem is solved using alternatively the Cerlienco-Mureddu Correspondence, the Lex Game, the Gao-Rodrigues-Stroomer algorithm or the Lederer's algorithm. The second problem can be solved by an algorithm due to Lazard.
In section 3.2 we will deal exactly with Lazard's algorithm.
In section 3.3, we will give an overview of Lazard's structural theorem and of another result about factorization, named *Macaulay's Trick*.
In the fourth section we will explain the Axis of Evil algorithm in detail, and in section 3.5 we will summarize some results which can be considered as consequences of the Axis of Evil theorem.
Finally, in section 3.6 we will give a very detailed example of execution of the original Axis of Evil algorithm 5.

3.2 Considerations on the monomial basis and Lazard's algorithm.

In this section, we make some remarks on the behaviour of the monomial basis $\mathsf{G}(I)$ of a zerodimensional ideal I.
First of all, we deal with the most efficient way to compute it from the Groebner escalier $\mathsf{N}(I)$ of I, namely *Lazard's algorithm*.
After that, we will study the structure of $\mathsf{G}(I)$ degree by degree, defining the concept of *natural expansion*.
We will exploit the diagrams defined in 1.5 in order to represent and distinguish the terms in $\mathsf{N}(I)$ and $\mathsf{G}(I)$.
Lazard's algorithm ([36, 79]) is a very simple but powerful tool in order to study zerodimensional ideals.
It has been developed in [36], actually being a part of FGLM algorithm.
The aim of Lazard's algorithm is to compute the monomial basis $\mathsf{G}(I)$ of a zerodimensional ideal $I \lhd \mathbf{k}[x_1, ..., x_n]$ having, as input, only the Groebner escalier $\mathsf{N}(I)$. This algorithm is *iterative* on the terms in $\mathsf{N}(I) = \{\tau_1, ..., \tau_s\}$. Start with $|\mathsf{N}(I)| = 1$, namely $\mathsf{N}(I) = \{1\}$[2]. Then the monomial basis is $\mathsf{G}(I) = \{x_1, ..., x_n\}$, since for each $j \in \{1, ..., n\}$ the only existing predecessor of x_j is $1 \in \mathsf{N}(I)$, while no other term σ can belong to $\mathsf{G}(I)$, being multiple of at

[2]The only order ideal with cardinality one is exactly the singleton $\{1\}$, by the definition of order ideal itself.

least a variable.
Set also $L = [x_1, ..., x_n]$ i.e. a list containing the products $1 \cdot x_j$, for $j = 1, ..., n$.
The above steps constitute the basis for our procedure.
Let $|\mathsf{N}(I)| > 1$, $\mathsf{G}_{i-1} = \{\tau'_1, ..., \tau'_h\}$ be the monomial basis associated to the order ideal $\mathsf{N}_{i-1} = \{1, \tau_2, ..., \tau_{i-1}\}$, $i \leq S$ and L the list (ordered w.r.t. lex) containing the products of the form $\tau_k x_j$, for $k = 1, ..., i-1$, $j = 1, ..., n$, with $\tau_k x_j \notin \mathsf{N}_{i-1}$. We do not allow repetitions in L, so if $\sigma = x_{j_0}\tau_{j_0} = x_{j_1}\tau_{j_1}$, σ is reported only once in L, but it is marked with a number, i.e. the number of times it has been computed.
Consider then $\tau_i \in \mathsf{N}(I)$; in order to compute the monomial basis associated to $\mathsf{N}_i = \{\tau_1, ..., \tau_i\}$, Lazard's algorithm performs the the steps displayed below on τ_i.

- remove τ_i both from L and from G_{i-1};
- Computes all the products $\sigma_{j,i} = x_j\tau_i$, for each $j = 1, ..., n$.
- Inserts each $\sigma_{j,i}$ in L. For each $\sigma_{\overline{j},i}$ already appearing in L, the algorithm marks the number of times it has been computed and selected for insertion.
- All the terms appearing in L, marked exactly with the number of their variables, are the elements of G_i, the monomial basis associated to N_i.

Remark 3.2.1 ([36])**.** We study now Lazard algorithm from the efficiency point of view. As proved in [36], its complexity is $O(n^2s^2)$, where $s = |\mathsf{N}(I)|$ and n is the number of variables in the given polynomial ring. In the same paper, the authors remarked also that, with a more efficient implementation, involving priority queues, the complexity of the algorithm can be improved to $O(n^2 s \log(ns))$.

We give now a simple example of execution for Lazard's algorithm.

Example 3.2.2. Consider the order ideal $\mathsf{N}(I) = \{1, x_1, x_2, x_1x_2, x_3\} \subseteq \mathbf{k}[x_1, x_2, x_3]$. In order to compute $\mathsf{G}(I)$ we proceed term by term as displayed in in the list below.

1: this is the base case, so we get $L = [x_1, x_2, x_3]$ and all the terms coincide with the monomial basis associated to $\{1\}$.

x_1: we get $L = [x_1^2, x_2, x_1x_2, x_3, x_1x_3]$. All terms appear only once, two of them containing two variables, namely x_1x_2, x_1x_3, do not belong to the monomial basis associated to $\{1, x_1\}$.

x_2: we get $L = [x_1^2, \underbrace{x_1x_2}_{2 \text{ times}}, x_2^2, x_3, x_1x_3, x_2x_3]$. This time, x_1x_2 turns out to be in the monomial basis, since it appears twice and it contains two variables, which is not the case for x_1x_3, x_2x_3.

x_1x_2: here, we obtain $L = [x_1^2, x_1^2x_2, x_2^2, x_1x_2^2, x_3, x_1x_3, x_2x_3, x_1x_2x_3]$. All the terms appear once, so we remove all the ones containing more than one variable.

x_3: for x_3, finally, we get $L = [x_1^2, x_1^2x_2, x_2^2, x_1x_2^2, \underbrace{x_1x_3}_{\text{2 times}}, \underbrace{x_2x_3}_{\text{2 times}}, x_1x_2x_3, x_3^2]$. Here, we have to remove $x_1^2x_2, x_1x_2^2, x_1x_2x_3$.

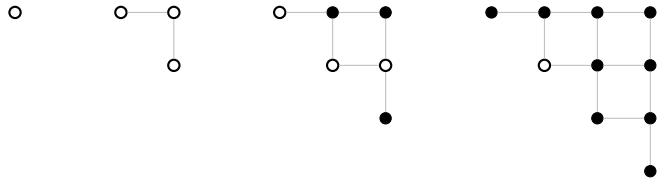
The monomial basis for $\mathsf{N}(I) = \{1, x_1, x_2, x_1x_2, x_3\}$ is then $\mathsf{G}(I) = \{x_1^2, x_2^2, x_1x_3, x_2x_3, x_3^2\}$.

Dealing with the monomial basis, we also study its behaviour degree by degree, representing it in a very concrete way. This goal can be achieved defining the *natural expansion*.

Definition 3.2.3. Let $H \subseteq \mathcal{T}_j$ for some $j \in \mathbb{N}^*$ we set $C^{(0)}(H) := H$ and, for all $l \in \mathbb{N}^*$ $C^{(l)}(H) = \{\tau \in \mathcal{T}_l, \exists \sigma \in H, \sigma \mid \tau\}$.
The set $C^{(l)}(H)$ is the *natural expansion* of H at degree l.

Given then a finite order ideal N, we arrange it by degree, obtaining N_0, N_1, $\cdots$ N_h, where h is the maximal degree of terms belonging to N.
The monomial basis G associated to such an N can have at most degree $h+1$.
As a matter of fact, if $\tau \in \mathsf{G}$ with $deg(\tau) = d > h+1$ its predecessors will belong to N and then we have terms of degree $d-1 \geq h+1$ in the order ideal, what is impossible by hypothesis.

Example 3.2.4. There are situations in which N contains monomials of degree at most h, but also the minimal basis shares the same property.
Take $I = (x^3, y^2, z^2, xy) \triangleleft \mathbf{k}[x, y, z]$, whose Groebner escalier is:
$\mathsf{N}_0 = \{1\}$
$\mathsf{N}_1 = \{x, y, z\}$
$\mathsf{N}_2 = \{yz, xz, x^2\}$
$\mathsf{N}_3 = \{x^2z\}$:



The monomial basis does not contain elements of degree 4.

We call G_i the i-degree part of the monomial basis $\mathsf{G}(I)$.

Lemma 3.2.5. For all $i = 0, ..., h+1$

$$\mathcal{T}_i \setminus (\mathsf{N}_i \cup \bigcup_{j=1}^{i-1} C^{(i)}(\mathsf{G}_j)) = \mathsf{G}_i.$$

Proof: The inclusion $\mathcal{T}_i \setminus (\mathsf{N}_i \cup \bigcup_{j=1}^{i-1} C^{(i)}(\mathsf{G}_j)) \supseteq \mathsf{G}_i$ is trivial, so we only prove the converse, $\mathcal{T}_i \setminus (\mathsf{N}_i \cup \bigcup_{j=1}^{i-1} C^{(i)}(\mathsf{G}_j)) \subseteq \mathsf{G}_i$.
Consider $\tau \in \mathcal{T}_i \setminus (\mathsf{N}_i \cup \bigcup_{j=1}^{i-1} C^{(i)}(\mathsf{G}_j))$. Clearly $\tau \in I$.
Let σ the h-th predecessor of τ; if $\sigma \in I$, $\exists \theta \in \mathsf{G}(I)$ with $\sigma = \theta \cdot \mu$ for a suitable $\mu \in \mathcal{T}$.
Then $\tau = \theta \cdot \mu \cdot x_h$ i.e. $\tau \in \bigcup_{j=1}^{i-1} C^{(i)}(\mathsf{G}_j)$. $\diamond$

Example 3.2.6. For $I = (x^3, y^2, z^2, xy) \lhd \mathbf{k}[x, y, z]$, (see example 3.2.4) we have $G_0 = G_1 = \emptyset$, $G_2 = \mathcal{T}_2 \setminus \{yz, xz, x^2\} = \{y^2, z^2, xy\}$ and $G_3 = \mathcal{T}_3 \setminus (\{xy^2, xyz, x^2y, xz^2, yz^2, z^3, y^3, y^2z\} \cup \{x^2z\}) = \{x^3\}$.

3.3 Macaulay Trick and Lazard Structural Theorem.

In this section, we focus on two famous results on factorized Groebner bases, namely *Macaulay Trick* and *Lazard structural theorem*.
We start dealing with the setting examined by Macaulay, studying a way to solve the problem below.

Problem 3.3.1. Given a finite set of terms $\{\tau_1, ..., \tau_r\} \subset \mathcal{T}$ and a term order $<$ on $\mathcal{T}$, construct a set of polynomials $\{g_1, ..., g_r\} \subset P$ such that:

- for each $i \in \{1, ..., r\}$, $\mathsf{T}(g_i) = \tau_i$;
- $\mathcal{G} := \{g_1, ..., g_r\}$ is a Groebner basis for the ideal $I = (\mathcal{G})$, that is

$$\mathsf{T}(I) = \mathsf{T}(\mathcal{G}) = (\tau_1, ..., \tau_r).$$

Description 3.3.2. In order to look for a solution, we first construct a finite sequence

$$M := [\sigma_1, ..., \sigma_s] \subseteq \mathcal{T}$$

satisfying:

a. for each i, $1 \leq i \leq r$ exists a subset $J_i \subset \{1, ..., s\}$ such that $\tau_i = \prod_{l \in J_i} \sigma_l$;

b. for each i, j $1 \leq i < j \leq r$, $lcm(\tau_i, \tau_j) = \prod_{l \in J_i \cup J_j} \sigma_l$

Remark 3.3.3. We point out that, by definition, M is a *finite sequence* and not a set, so *repetitions* among the elements appearing in M *are allowed*.

Example 3.3.4. For the terms $\tau_1 := x^2$ and $\tau_2 := xy$ in $\mathbf{k}[x, y]$, we get

$$\sigma_1 := \sigma_2 := x,\ \sigma_3 := y$$

and

$$J_1 := \{1, 2\},\ J_2 := \{1, 3\}.$$

Remark 3.3.5. The finite sequence satisfying conditions $a.$ and $b.$ *is not unique*.
Given $\{\tau_1, ..., \tau_r\} \subset \mathcal{T}$, more than one sequence can produce the required result, as shown in the following example.

Example 3.3.6. For the terms $\tau_1 := x^4,\ \tau_2 := x^3y^3$ in $\mathbf{k}[x, y]$ we can consider first the sequence

$$M_1 := [x, x^3, y^3].$$

Using the list M_1, we have $J_1 = \{1, 2\},\ J_2 = \{2, 3\}$.
Indeed, we have $x \cdot x^3 = x^4 = \tau_1$, $x^3 \cdot y^3 = x^3y^3 = \tau_2$ and it holds

$$lcm(x^4, x^3y^3) = x^4y^3 = x \cdot x^3 \cdot y^3 = \prod_{l \in J_1 \cup J_2} \sigma_l.$$

Moreover, we notice that $GCD(x^4, x^3y^3) = x^3 = \prod_{l \in J_1 \cap J_2} \sigma_l$.
However, M_1 is not the unique sequence compatible with conditions $a.$ and $b.$
Consider indeed the sequence

$$M_2 := [x, x, x, x, y, y, y].$$

For M_2, we get $J_1 = \{1, 2, 3, 4\}$, $J_2 = \{1, 2, 3, 5, 6, 7\}$.
Indeed, $x \cdot x \cdot x \cdot x = x^4 = \tau_1$ and $x \cdot x \cdot x \cdot y \cdot y \cdot y = x^3y^3 = \tau_2$.
Moreover, it holds $lcm(x^4, x^3y^3) = x^4y^3 = x \cdot x \cdot x \cdot x \cdot y \cdot y \cdot y$.

We describe now an algorithmic method in order to compute concretely a sequence of the required shape.
Given a set of terms $\{\tau_1, ..., \tau_r\} \subset \mathcal{T}$, defined as $\tau_1 := x_1^{\alpha_{1,1}} \cdots x_n^{\alpha_{1,n}}, ..., \tau_r := x_1^{\alpha_{r,1}} \cdots x_n^{\alpha_{r,n}}$.
For this set, we can consider the following sequence, only composed by *single variables*:

$$M := [x_1, ..., x_1, x_2, ..., x_2, ..., x_n, ..., x_n],$$

where for each $1 \leq h \leq n$, x_h appears exactly $\alpha_h := \max\{\alpha_{1,h}, ..., \alpha_{r,h}\}$ times so that $|M| = \sum_{h=1}^{n} \alpha_h$ and we number the elements of M from 1 to $|M|$.

Given any term $\tau_i = x_1^{\alpha_{i,1}} \cdots x_n^{\alpha_{i,n}}$ in the given set, the associated J_i can be computed as follows

$$J_i = \{1, ..., \alpha_{i,1}, \alpha_i + 1, ..., \alpha_1 + \alpha_{i,2} - 1, ..., \alpha_1 + ... + \alpha_{n-1} + 1, ..., \alpha_1 + ... + \alpha_{n-1} + \alpha_{i,n} - 1\}.$$

We show a simple example of the above construction.

Example 3.3.7. If the given terms are $\tau_1 = x_1^4$, $\tau_2 = x_1^3$, $\tau_3 = x_2$, $\tau_4 = x_1^2$, we consider the sequence

$$M = [x_1, x_1, x_1, x_1, x_2],$$

labelling its elements as $L = [1, 2, 3, 4, 5]$. The term $\tau_1 = x_1^4$ contains only x_1, with exponent 4, so we get $J_1 = \{1, 2, 3, 4\}$.
For $\tau_2 = x_1^3$, we take the first three numbers, labelling copies of x_1, so $J_2 = \{1, 2, 3\}$.
Since $\tau_3 = x_2$, we get $J_3 = \{5\}$ and finally, for $\tau_4 = x_1^2$, we obtain $J_4 = \{1, 2\}$.
The crucial fact is to take *the first numbers of the list L for the variables*. Indeed, if we take $J_4' = \{3, 4\}$ instead of J_4, we get $lcm(\tau_2, \tau_4) = x_1^4$, since we have to derive it from $J_2 \cup J_4' = \{1, 2, 3, 4\}$, but this is clearly false.

Clearly, condition a. of description 3.3.2 is fulfilled: $\tau_i = \prod_{l \in J_i} \sigma_l$.
On the other hand, suppose to consider the union $J_i \cup J_j$ of two sets obtained from a finite sequence as above. Such operation corresponds to take the common and non common factors of the associated terms τ_i, τ_j, raised to the maximal exponents they appear with. It exactly means computing the least common multiple between τ_i and τ_j:

$$lcm(\tau_i, \tau_j) = \prod_{l \in J_i \cup J_j} \sigma_l.$$

For each $l,\ 1 \leq l \leq s$ we choose a polynomial $h_l \in P = \mathbf{k}[x_1, ..., x_n]$ such that $\mathsf{T}(h_l) < \sigma_l$ and we define:

$$\gamma_l := \sigma_l - h_l,\ \forall l,\ 1 \leq l \leq s;$$
$$g_i := \prod_{l \in J_i} \gamma_l,\ \forall i,\ 1 \leq i \leq r.$$

It holds $\mathsf{T}(g_i) = \prod_{l \in J_i} \sigma_l$.
With the above notation, for each couple of indices $i, j,\ 1 \leq i < j \leq r$, denoted

$$\mathsf{T}(i, j) = lcm(\mathsf{T}(g_i), \mathsf{T}(g_j)) = lcm(\tau_i, \tau_j),$$

we choose $t_{i,j}, t_{j,i} \in \mathcal{T}$ defined as

$$t_{i,j}\mathsf{T}(g_i) = t_{j,i}\mathsf{T}(g_j).$$

Assuming

$$lcm(\tau_i, \tau_j) = \Big(\frac{\prod_{l\in J_i}\sigma_l \cdot \prod_{l\in J_j}\sigma_l}{\prod_{l\in J_i\cap J_j}\sigma_l}\Big)$$

$$= \prod_{l\in J_i\cup J_j}\sigma_l = \prod_{l\in J_i}\sigma_l \cdot \prod_{l\in J_j\setminus J_i}\sigma_l = \prod_{l\in J_j}\sigma_l \cdot \prod_{l\in J_i\setminus J_j}\sigma_l,$$

it clearly holds

$$t_{i,j} := \prod_{l\in J_j\setminus J_i}\sigma_l,$$

$$t_{j,i} := \prod_{l\in J_i\setminus J_j}\sigma_l.$$

Proposition 3.3.8. With the above notation, the set $\mathcal{G} := \{g_1, ..., g_r\}$ is a Groebner basis.

Proof: We prove that, considered two arbitrary i, j, $1 \le i < j \le r$, the S-polynomial $S(i, j)$ has a Groebner representation.
For this purpose, we define

$$\phi_{i,j} := \Big(\prod_{l\in J_j\setminus J_i}\gamma_l\Big) - t_{i,j};$$

$$\phi_{j,i} := \Big(\prod_{l\in J_i\setminus J_j}\gamma_l\Big) - t_{j,i};$$

We know that

$$t_{i,j} = T\Big(\prod_{l\in J_j\setminus J_i}\gamma_l\Big)$$

$$t_{j,i} = T\Big(\prod_{l\in J_i\setminus J_j}\gamma_l\Big)$$

and, since in $\phi_{i,j}, \phi_{j,i}$ we subtract to the above products exactly the leading terms, we can affirm that $\mathsf{T}(\phi_{i,j}) < t_{i,j}$ and $\mathsf{T}(\phi_{j,i}) < t_{j,i}$.
We prove then that the required representation is

$$S(i, j) = -\phi_{i,j}g_i + \phi_{j,i}g_j.$$

In effect this is true since, by the properties of union

$$0 = -\prod_{l\in J_i\cup J_j}\gamma_l + \prod_{l\in J_j\cup J_i}\gamma_l =$$

so, manipulating the formula, we get

$$= -\prod_{l\in J_i}\gamma_l\Big(\prod_{l\in J_j\setminus J_i}\gamma_l\Big) + \prod_{l\in J_i}\gamma_l\Big(\prod_{l\in J_i\setminus J_j}\gamma_l\Big) =$$

$$= -g_i\Big(\prod_{l\in J_j\setminus J_i}\gamma_l\Big) + g_j\Big(\prod_{l\in J_i\setminus J_j}\gamma_l\Big) =$$

but, by definition of $\phi_{i,j}, \phi_{j,i}$,

$$= -(\phi_{i,j} + t_{i,j})g_i + (\phi_{j,i} + t_{j,i})g_j = -\phi_{i,j}g_i + \phi_{j,i}g_j - (t_{i,j}g_i - t_{j,i}g_j) =$$

and by definition of S-polynomial

$$= -\phi_{i,j}g_i + \phi_{j,i}g_j - S(i,j).$$

For being effectively a Groebner representation, the condition on the leading terms must be fulfilled.
Anyway, the following relations imply directly that condition:

$$\mathsf{T}(\phi_{i,j}g_i) < t_{i,j}\mathsf{T}(g_i) = lcm(\mathsf{T}(g_i), \mathsf{T}(g_j)) = t_{j,i}\mathsf{T}(g_j) > \mathsf{T}(\phi_{j,i}g_j).$$

$\diamond$

This way, we have then solved the problem 3.3.1.
We switch now to a new problem, solved by Macaulay.
Consider a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\} \subset \mathbf{k}^n$, with $P_i := (a_{i1}, ..., a_{in})$ and set the following notation:

- $\forall i$, $l_i \in \mathcal{P}^* = Hom_k(\mathcal{P}, k)$ is the linear functional, operating the "evaluation" in the associated point:
$$l_i(f) = f(a_{i1}, ..., a_{in})\ \forall f(x_1, ..., x_n) \in \mathcal{P};$$
- $L(\mathbf{X}) := Span_k(\{l_i,\ 1 \le i \le S\}) \subset \mathcal{P}^*$
- $I(\mathbf{X}) := \{f \in \mathcal{P} : f(P_i) = 0,\ \forall i\} = \mathfrak{P}(L(\mathbf{X}))$, the ideal of points for X.

Under the above notation, we can present the following result by Macaulay (see [68]).
Let $\mathsf{N} \subset \mathcal{T}$ be a finite order ideal.
Let $J := \mathcal{T} \setminus \mathsf{N}$ be the associated semigroup ideal and $\mathsf{G}(J) := \{\tau_1, ..., \tau_r\}$, with $\tau_l = x_1^{\alpha_{1l}} \cdots x_n^{\alpha_{nl}}$ for each l.
Since N is a finite set, for each $i \in \{1, ..., n\}$ we need to have a $d_i \in \mathbb{N}$ such that

$$x_i^{d_i} \in \mathsf{G}(J)$$

and, moreover,

$$\alpha_{il} \le d_i\ \forall l.$$

Example 3.3.9. If we consider the polynomial ring $\mathbf{k}[x_1, x_2]$ and we take the finite order ideal $\mathsf{N} = \{1, x_1, x_2, x_2^2, x_1x_2\}$, the associated monomial basis is $\mathsf{G} = \{x_2^3, x_1^2, x_1x_2^2\}$ and, in this case, $d_1 = 2$, $d_2 = 3$.

For each $i, j, e,\ j \neq e$ we choose the elements

$$a_{ij} \in k,\ 1 \leq i \leq n,\ 0 \leq j \leq d_i :\ a_{ij} \neq a_{ie}$$

and, for each l, $1 \leq l \leq r$

$$g_l := \prod_{i=1}^{n} \prod_{j=0}^{\alpha_{il}-1} (x_i - a_{ij}),$$

for which, trivially $\mathsf{T}(g_l) = \tau_l$ holds.
We associate to each term $t = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{N}$ an affine point

$$a(t) := (a_{1\alpha_1}, ..., a_{n\alpha_n}) \in \mathbf{k}^n$$

and we set

$$\mathbf{X} := \{a(t) : t \in N\}.$$

We obtain then

Corollary 3.3.10. With the above notation, for each degree-compatible term order, we have:

1. $\mathsf{N} = \mathsf{N}(\mathsf{I}(\mathbf{X}))$;
2. $\mathcal{G}(I(\mathbf{X})) := \{g_1, ..., g_r\}$ is the reduced Groebner basis of $I(\mathbf{X})$.

Proof: First of all, we notice that we are under the hypotheses of proposition 3.3.8.
Indeed, the chosen numbers a_{ij} play the role of the elements h_l defined above (we consider the list containing all the terms σ_l constructed as explained before: a_{ij} is the element related to the i-th variable and the $j+1$-th exponent for x_i).
Moreover, the product constituting the polynomials g_i's, for $i = 1, ..., n$ and $j = 0, ..., \alpha_{il} - 1$ coincides with the product of the γ_l with $l \in J_i$.
With that, the set $\mathcal{G} = \{g_1, ..., g_r\}$ represents a Groebner basis for the ideal $J = (g_1, ..., g_r)$ and N is the Groebner escalier for the ideal whose Groebner basis is $\mathcal{G}$.
Since by construction, all the polynomials vanish over X, we have $J \subseteq I(\mathbf{X})$.
Moreover, by the relations

$$mult(J) = |\mathsf{N}| = |\mathbf{X}| = mult(I(\mathbf{X})),$$

we can conclude $J = I(\mathbf{X})$.
We can say that $\mathcal{G} = \mathcal{G}(J) = \mathcal{G}(I(\mathbf{X}))$ is a Groebner basis of $I(\mathbf{X})$ and $\mathsf{N} = \mathsf{N}(I(\mathbf{X}))$.
Such a basis is also the reduced one because:

- it is composed by monic polynomials;
- G is minimal;
- the polynomials g_i, $i = 1, ..., r$, have the form $\mathsf{T}(g_i) - Can(\mathsf{T}(g_i), i)$, since $Supp(g_i) \setminus \{\mathsf{T}(g_i)\} \subseteq \mathsf{N}$. Actually, these terms divide $\mathsf{T}(g_i)$ by construction. Moreover, the polynomials g_i belong to the ideal.

$\diamond$

Let us consider now a very simple example.

Example 3.3.11. In the polynomial ring $\mathbf{k}[x, y]$, we consider the finite order ideal $\mathsf{N} = \{1, x, y\}$. In our notation the monomial basis turns out to be $\mathsf{G} = \{x^2, xy, y^2\}$.
The pure powers of x, y in G[3] have to be raised to the exponents $d_1 = d_2 = 2$ in the above notation, since 2 is the minimal power of x, y in N. Indeed, no mixed products of the form $x^i y^j$ can have i or j greater than the value in the corresponding pure power, by minimality of G and $x^2, y^2 \in G$.
Fix the following values:
$a_{10} = 0, a_{11} = 1, a_{20} = 0, a_{21} = 1$, obtaining the points:
$a(1) = (0, 0)$;
$a(x) = (0, 1)$;
$a(y) = (1, 0)$,
i.e. $\mathbf{X} = \{(0, 0), (0, 1), (1, 0)\}$. The polynomials g_i, $i = 1, ..., 3$, will be

$$g_1 := x^2 - x;$$

$$g_2 := xy;$$

$$g_3 := y^2 - y,$$

and we have exactly $I(\mathbf{X}) = \{g_1, g_2, g_3\}$.

Given an arbitrary

$$\sigma = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \{x_j \tau / 1 \leq j \leq n,\ \tau \in \mathsf{N}\},$$

[3] They surely exist, since $|\mathsf{N}| < \infty$.

in the zerodimensional case, we have that $\alpha_i \leq d_i$, for each $i \in \{1, ..., n\}$, so it is natural to consider the following polynomials:

$$g_\sigma := \prod_{i=1}^{n} \prod_{j=0}^{\alpha_i - 1} (x_i - a_{ij}),\ \sigma = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \{x_j \tau / 1 \leq j \leq n,\ \tau \in N\}$$

and study their relations, leaning on the notation above.
First of all, we reorder the order ideal $\mathsf{N} := \{\tau_1, ..., \tau_S\}$ increasingly w.r.t the lexicographical order induced by $x_1 < ... < x_n$ and we set $a_i := a(t_i)$ in order to fix also an order both on $\mathbf{X}$ and $L(\mathbf{X})$.
Finally, we set $q_i := g_{\tau_i}$, for each $i \in \{1, ..., S\}$. It holds

Lemma 3.3.12. With the above notation, we get

1. $\mathcal{B}(I(\mathbf{X})) = \{g_\tau / \tau \in \mathsf{B}(I(\mathbf{X}))\}$;
2. $\mathcal{G}(I(\mathbf{X})) = \{g_\tau / \tau \in G(I(\mathbf{X}))\}$;
3. $q(\mathbf{X}) = \{q_i / 1 \leq i \leq S\}$.

Proof:

1. For this statement, we barely follow the line of 3.3.10: the polynomials belong to $I(\mathbf{X})$ and their leading terms are in the border set, while the other terms appearing in their support belong to the Groebner escalier;
2. it is 3.3.10;
3. we have to prove that the q_i's are triangular, i.e. $l_i(t_j) = 0$ for $i < j$.
   In our case, the functionals are the evaluations at points, so we need to prove that $g_{t_j}(a(t_i)) = 0, i < j$.
   By the ordering given to the terms, $\tau_i < \tau_j$. It means that, if $\tau_i = x_1^{\alpha_{i,1}} \cdots x_n^{\alpha_{i,n}}$ and $\tau_j = x_1^{\alpha_{j,1}} \cdots x_n^{\alpha_{j,n}}$, there exists $h \in \{1, ..., n\}$ such that $\alpha_{j,h} > \alpha_{i,h}$.
   For this reason, constructing g_{τ_j} we get a factor vanishing in a_{τ_i} and then we can conclude.

$\diamond$

If we deal with the polynomial ring in two variables $\mathbf{k}[x_1, x_2]$, the Groebner basis constructed via Macaulay's trick for an ideal I as before, is an example illustrating *Lazard structural theorem*.

This theorem describes the structure of a lexicographical minimal Groebner basis for an ideal $I \triangleleft \mathbf{k}[x_1, x_2]$.
The proof considers $\mathcal{P} = \mathbf{k}[x_1, x_2] = \mathbf{k}[x_1][x_2]$ and bases on the fact that $\mathbf{k}[x_1]$ is a Principal Ideal Domain (PID).
We can then extend it to the more general case $R[x]$, with R PID, to describe Groebner bases.
In order to understand the statement of Lazard structural theorem, we first recall the following definitions.

Definition 3.3.13. The *content* $r_f \in R$, with R PID, of a polynomial $f(x) \in R[x]$ is the GCD of its coefficients. A polynomial $f(x) \in R[x]$ is called *primitive* if $r_f = 1$.
The *primitive part* of $f(x) \in R[x]$ is the polynomial $p_0(x) \in R[x]$ such that $f(x) = r_f p_0(x)$.

We first prove the following

Proposition 3.3.14. Let R be a principal ideal ring and $I \triangleleft P := R[x]$ an ideal. Let $F := \{f_0, ..., f_s\}$ be a minimal Groebner basis of I, ordered so that

$$deg(f_0) \leq ... \leq deg(f_s)$$

and, for each i, denote by $c_i := Lc(f_i)$, $r_i \in R \setminus \{0\}$ and by $p_i \in P$ the content and the primitive part of f_i. We can further assume that such basis is reduced, in the sense that

$$f_i = \mathsf{M}(f_i) + Can(\mathsf{M}(f_i), F).$$

Then

1. $deg(f_0) < ... < deg(f_s)$;
2. for each $0 \leq i < s$ there is $G_{i+1} \in R$ such that $c_i = G_{i+1} c_{i+1}$
3. $G_{i+1} f_{i+1} \in (f_0, ..., f_i)$ for each $0 \leq i < s$.

Proof: Let us set $d(i) := deg(f_i)$ for each i. By hypothesis, we have $d(i) \leq d(i+1)$.
We prove, first of all, that the case $d(i) = d(i+1)$ cannot occur. Indeed, if $d(i) = d(i+1)$ we can define the element

$$h := b_i f_i + b_{i+1} f_{i+1} \in I,$$

where c, b_i, b_{i+1} belong to R and $b_i c_i + b_{i+1} c_{i+1} = c = GCD(c_i, c_{i+1})$, so that $cx^{d(i+1)} = \mathsf{M}(h) \in \mathsf{M}(I)$.
Since $\mathsf{M}(h) \in \mathsf{M}(I)$ there exists an index j with $\mathsf{M}(f_j) \mid \mathsf{M}(h) \mid \mathsf{M}(f_{i+1})$[4].

[4] The first divisibility relation comes from the fact that $\mathsf{M}(h) \in \mathsf{M}(I)$, while the second one is consequence of $\mathsf{T}(f_{i+1}) = x^{d(i+1)}$ and $c = GCD(c_i, c_{i+1}) | c_{i+1}$.

This chain of relations assures that $\mathsf{M}(f_j)|\mathsf{M}(f_{i+1})$ i.e. gives a divisibility relation between the leading terms of two elements in the basis. By the minimality, this is impossible, so $d(i) < d(i+1)$.
Both $x^{d(i+1)d(i)}f_i$ and f_{i+1} are in the ideal and have degree $d(i+1)$; then for $c, b_i, b_{i+1} \in R$ such that $b_ic_i + b_{i+1}c_{i+1} = c = GCD(c_i, c_{i+1})$, $h := b_i x^{d(i+1)-d(i)}f_i + b_{i+1}f_{i+1} \in I$, so that $cx^{d(i+1)} = \mathsf{M}(h) \in \mathsf{M}(I)$ and $\mathsf{M}(f_j) \mid \mathsf{M}(h)$ for some j. If $c_{i+1} \neq GCD(c_i, c_{i+1})$, then $j < i+1$ and $\mathsf{M}(f_j)|\mathsf{M}(f_{i+1})$, getting a contradiction. As a conclusion, $c_{i+1} \mid c_i$ for each i.
Since $G_{i+1}f_{i+1} - xd(i+1) - d(i)f_i$ is a polynomial of degree less than $d(i+1)$ reducing to 0 w.r.t. the Groebner basis, so $G_{i+1}f_{i+1} \in (f_0, ..., f_i)$. $\diamond$

Theorem 3.3.15. With the same notation, if moreover R is a domain, denoting by $p := p_0$ the primitive part of f_0 and $G_{s+1} := r_s \in R \setminus \{0\}$ the content of of f_s, then for each i, $0 \leq i < s$ there is $H_{i+1} \in P$, $d(i) := deg(H_i)$ such that

- $f_0 = pG_1 \cdots G_{s+1}$;
- $f_j = pH_jG_{j+1} \cdots G_{s+1}$, $1 \leq j \leq s$

and

1. $r_i = G_{i+1} \cdots G_s$
2. $Lc(H_i) = 1$ for each i
3. $d(1) < ... < d(s)$;
4. for each i, we have $H_{i+1} \in (G_1 \cdots G_i, H_1G_2 \cdots G_i, ..., H_{i-1}G_i, H_i)$;

Proof: Let p and G_{s+1} be, respectively, the primitive part and the content of $GCD(f_0, ..., f_k)$ in $R[x]$; a set $\{g_0, ..., g_s\}$ is a minimal Groebner basis if and only if so is for $\{gg_0, ..., gg_s\}$, we can divide by pG_{s+1} and assume that $p = G_{s+1} = 1$ and $GCD(f_0, ..., f_s) = 1$. Under this assumption, $G_{i+1}f_{i+1} \in (f_0, ..., f_i)$ for each i, $0 \leq i < k$ so, inductively, we have

- $p_0|f_j$, $\forall j \leq i \Rightarrow p_0|f_j$, $\forall j \leq i+1$;
- $c_i|f_j$, $\forall j \leq i \Rightarrow c_i = G_{i+1}c_{i+1}|G_{i+1}f_{i+1}$, $\forall j \leq i+1 \Rightarrow$
  $\Rightarrow c_{i+1}|f_j$, $\forall j \leq i+1$.

Therefore, $GCD(f_0, ..., f_s) = 1$ gives that $p_0 = c_s = 1$ and each c_i verifies $c_i|f_i$, so it coincides with r_i.
By induction, we have

$$lc(p)r_i = c_i = G_{i+1}c_{i+1} = lc(p)G_{i+1}r_{i+1} = lc(p)G_{i+1} \cdots G_s.$$

Setting $H_i := \frac{f_i}{c_i}$ for each i, we obtain $lc(H_i) = 1$, $d(i) + deg(p) = deg(f_i)$ and finally, we point out that $G_{i+1}f_{i+1} \in (f_0, ..., f_i)$: dividing $G_{i+1} \cdots G_s$ we can conclude. $\diamond$

Example 3.3.16. Consider again example 3.3.11, and set the lexicographical order, $x < y$.
The Groebner basis is $\{x^2 - x, xy, y^2 - y\}$, which is ordered as in 3.3.15.
We have $p = 1$, $G_1 = x_1$, $G_2 = x$, $G_3 = 1$, $H_1 = y$, $H_2 = y^2 - y$, $d(1) = 1 < d(2) = 2$ and $H_2 \in (G_1, H_1)$.

In the next example, we apply Macaulay trick, showing a relationship with Lazard structural theorem.

Example 3.3.17. Consider the polynomial ring in three variables $\mathcal{P} = \mathbf{k}[x_1, x_2, x_3]$, the associated set of terms $\mathcal{T}$ and the lexicographical order induced by $x_1 < x_2 < x_3$.
Moreover, consider the order ideal

$$\mathsf{N} := \{1, x_1, x_1^2, x_2, x_1x_2, x_1^2x_2, x_2^2, x_1x_2^2, x_2^3, x_3, x_1x_3, x_1^2x_3, x_2x_3, x_2^2x_3, x_2^3x_3, x_3^2\}.$$

For each couple of indices i, j, we choose $a_{ij} = j$ and we consider the terms

$$\sigma \in (\{1\} \cup \{x_j\tau / 1 \leq j \leq n, \tau \in \mathsf{N}\}).$$

We will get:

1 : is a term in the order ideal N: $t_1 = 1 \in \mathsf{N}$. The corresponding point is $a(1) = (a_{10}, a_{20}, a_{30}) = (0, 0, 0) \in \mathbf{k}^3$ and we have $g_1 = q_1 = 1 \in q(\mathbf{X})$.

x_1 : $t_2 = x_1 \in \mathsf{N}$, $a(x_1) = (1, 0, 0)$, so $q_2 = g_{t_2} = g_{x_1} = x_1 \in q(\mathbf{X})$.

x_1^2 : $t_3 = x_1^2 \in \mathsf{N}$, $a(x_1^2) = (2, 0, 0)$, so $q_3 = g_{t_3} = g_{x_1^2} = x_1(x_1 - 1) \in q(\mathbf{X})$.

x_1^3 : $x_1^3 \notin \mathsf{N}$, and it is the product by x_1 of a term in N. Actually $x_1^3 \in \mathsf{G}$ (*all* the predecessors belong to N). Finally $g_{x_1^3} = x_1(x_1 - 1)(x_1 - 2) \in \mathcal{G}(I)$.

We proceed similarly:

x_2 : $\tau_4 = x_2 \in \mathsf{N}$, $a(x_2) = (0, 1, 0)$, $q_4 = g_{\tau_4} = g_{x_2} = x_2 \in q(\mathbf{X})$.

x_1x_2 : $\tau_5 = x_1x_2 \in \mathsf{N}$, $a(x_1x_2) = (1, 1, 0)$, $q_5 = g_{\tau_5} = g_{x_1x_2} = x_1x_2 \in q(\mathbf{X})$.

$x_1^2x_2$: $\tau_6 = x_1^2x_2 \in \mathsf{N}$, $a(x_1^2x_2) = (2, 1, 0)$, $q_6 = g_{\tau_6} = g_{x_1^2x_2} = x_1(x_1 - 1)x_2 \in q(\mathbf{X})$.

$x_1^3x_2:\ x_1^3x_2 \in B$ (caveat lector: $x_1^3 \notin \mathsf{N}$!), $g_{x_1^3x_2} = x_2(x_1-1)(x_1-2) \in \mathcal{B}(I)$.

$x_2^2:\ \tau_7 = x_2^2 \in \mathsf{N}$, $a(x_2^2) = (0,2,0)$, $q_7 = g_{\tau_7} = g_{x_2^2} = x_2(x_2-1) \in q(\mathbf{X})$.

$x_1x_2^2:\ \tau_8 = x_1x_2^2 \in \mathsf{N}$, $a(x_1x_2^2) = (1,2,0)$, $q_8 = g_{\tau_8} = g_{x_1x_2^2} = x_1x_2(x_2-1) \in q(\mathbf{X})$.

$x_1^2x_2^2:\ x_1^2x_2^2 \in \mathsf{G}$, $g_{x_1^2x_2^2} = x_1(x_1-1)x_2(x_2-1) \in \mathcal{G}(I)$.

$x_2^3:\ \tau_9 = x_2^3 \in \mathsf{N}$, $a(x_2^3) = (0,3,0)$, $q_9 = g_{\tau_9} = g_{x_2^3} = x_2(x_2-1)(x_2-2) \in q(\mathbf{X})$.

$x_1x_2^3:\ \tau_{10} = x_1x_2^3 \in \mathsf{N}$, $a(x_1x_2^3) = (1,3,0)$, $q_{10} = g_{\tau_{10}} = g_{x_1x_2^3} = x_1x_2(x_2-1)(x_2-2) \in q(\mathbf{X})$.

$x_1^2x_2^3:\ x_1^2x_2^3 \in B$, $g_{x_1^2x_2^3} = x_1(x_1-1)x_2(x_2-1)(x_2-2) \in \mathcal{B}(I)$.

$x_2^4:\ x_2^4 \in \mathsf{G}$, $g_{x_2^4} = x_2(x_2-1)(x_2-2)(x_2-3) \in \mathcal{G}(I)$.

$x_1x_2^4:\ x_1x_2^4 \in B$, $g_{x_1x_2^4} = x_1x_2(x_2-1)(x_2-2)(x_2-3) \in \mathcal{B}(I)$.

$x_3:\ \tau_{11} = x_3 \in \mathsf{N}$, $a(x_3) = (0,0,1)$, $q_{11} = g_{\tau_{11}} = g_{x_3} = x_3 \in q(\mathbf{X})$.

$x_1x_3:\ \tau_{12} = x_1x_3 \in \mathsf{N}$, $a(x_1x_3) = (1,0,1)$, $q_{12} = g_{\tau_{12}} = g_{x_1x_3} = x_1x_3 \in q(\mathbf{X})$.

$x_1^2x_3:\ \tau_{13} = x_1^2x_3 \in \mathsf{N}$, $a(x_1^2x_3) = (2,0,1)$, $q_{13} = g_{\tau_{13}} = g_{x_1^2x_3} = x_1(x_1-1)x_3 \in q(\mathbf{X})$.

$x_1^3x_3:\ x_1^3x_3 \in B$, $g_{x_1^3x_3} = x_1(x_1-1)(x_1-2)x_3 \in \mathcal{B}(I)$.

$x_2x_3:\ \tau_{14} = x_2x_3 \in \mathsf{N}$, $a(x_2x_3) = (0,1,1)$, $q_{14} = g_{\tau_{14}} = g_{x_2x_3} = x_2x_3 \in q(\mathbf{X})$.

$x_1x_2x_3:\ x_1x_2x_3 \in \mathsf{G}$, $g_{x_1x_2x_3} = x_1x_2x_3 \in \mathcal{G}(I)$.

$x_1x_2^4:\ x_1x_2^4 \in B$, $g_{x_1x_2^4} = x_1x_2(x_2-1)(x_2-2)(x_2-3) \in \mathcal{B}(I)$.

$x_1^2x_2x_3:\ x_1^2x_2x_3 \in B$, $g_{x_1^2x_2x_3} = x_1(x_1-1)x_2x_3 \in \mathcal{B}(I)$.

$x_2^2x_3:\ \tau_{15} = x_2^2x_3 \in \mathsf{N}$, $a(x_2^2x_3) = (0,2,1)$, $g_{x_2^2x_3} = g_{15} = q_{15} = x_2(x_2-1)x_3 \in q(\mathbf{X})$.

$x_1x_2^2x_3:\ x_1x_2^2x_3 \in B$, $g_{x_1x_2^2x_3} = x_1x_2(x_2-1)x_3 \in \mathcal{B}(I)$.

$x_2^3x_3:\ \tau_{16} = x_2^3x_3 \in \mathsf{N}$, $a(x_2^3x_3) = (0,3,1)$, $g_{x_2^3x_3} = g_{16} = q_{16} = x_2(x_2-1)(x_2-2)x_3 \in q(\mathbf{X})$.

$x_1x_2^3x_3:\ x_1x_2^3x_3 \in B$, $g_{x_1x_2^3x_3} = x_1x_2(x_2-1)(x_2-2)x_3 \in \mathcal{B}(I)$.

$x_2^4x_3:\ x_2^4x_3 \in B$, $g_{x_2^4x_3} = x_2(x_2-1)(x_2-2)(x_2-3)x_3 \in \mathcal{B}(I)$.

$x_3^2:\ \tau_{17} = x_3^2 \in \mathsf{N}$, $a(x_3^2) = (0,0,2)$, $g_{x_3^2} = g_{17} = q_{17} = x_3(x_3-1) \in q(\mathbf{X})$.

$x_1x_3^2:\ x_1x_3^2 \in \mathsf{G}$, $g_{x_1x_3^2} = x_1x_3(x_3-1) \in \mathcal{G}(I)$.

$x_1^2x_3^2:\ x_1^2x_3^2 \in B$, $g_{x_1^2x_3^2} = x_1(x_1-1)x_3(x_3-1) \in \mathcal{B}(I)$.

$x_2x_3^2:\ x_2x_3^2 \in \mathsf{G}$, $g_{x_2x_3^2} = x_2x_3(x_3-1) \in \mathcal{G}(I)$.

$x_2^2x_3^2:\ x_2^2x_3^2 \in B$, $g_{x_2^2x_3^2} = x_2(x_2-1)x_3(x_3-1) \in \mathcal{B}(I)$.

$x_2^3x_3^2:\ x_2^3x_3^2 \in B$, $g_{x_2^3x_3^2} = x_2(x_2-1)(x_2-2)x_3(x_3-1) \in \mathcal{B}(I)$.

$x_3^3:\ x_3^3 \in \mathsf{G}$, $g_{x_3^3} = x_3(x_3-1)(x_3-2) \in \mathcal{G}(I)$.

Now, we connect to Lazard Structural Theorem, considering the ideal $I \cap \mathbf{k}[x_1, x_2]$, whose Groebner basis is

$$\{g_{x_1^3}, g_{x_1^2x_2^2}, g_{x_2^4}\} = \{f_0, f_1, f_2\}.$$

The structure is exactly the one of the theorem

- $f_0 = x_1(x_1-1)(x_1-2) = G_1G_2$, dove $G_1 = (x_1-2), G_2 = x_1(x_1-1)$;
- $f_1 = x_1(x_1-1)x_2(x_2-1) = H_1G_2$, con $H_1 = x_2(x_2-1)$;
- $f_2 = x_2(x_2-1)(x_2-2)(x_2-3) = H_2$;

$p = G_3 = 1$, fulfilling the theorem.

3.4 The Axis of Evil algorithm.

For $I = \sqrt{I}$, the Axis of Evil Theorem by Marinari and Mora, somehow extends Lazard structural theorem 3.3.15 to the case of n variables, giving a remarkable improvement.
In this thesis, we give a constructive proof for

Theorem 3.4.1 (Marinari-Mora)**.** Consider a 0-dimensional radical ideal I. Denote by $\mathsf{N}(I)$ the associated Groebner escalier and $\mathsf{G}(I) = \{\tau_1, ..., \tau_r\} \subset \mathcal{T}$, $\tau_i := x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$ the monomial basis for the lexicographical initial ideal $In(I)$.
A combinatorial algorithm and interpolation provide polynomials

$$\gamma_{m\delta i} = x_m - g_{m\delta i}(x_1, ..., x_{m-1}),$$

for each $i \in \{1, ..., r\}$, $m \in \{1, ..., n\}$ and $\delta \in \{1, ..., d_{i,m}\}$ such that the products

$$f_i = \prod_m \prod_\delta \gamma_{m\delta i},\ i = 1, ..., r$$

form a minimal Groebner basis of I, with respect to the lexicographical order induced by $x_1 < ... < x_n$.

Clearly, for the polynomials f_i of theorem 3.4.1, we have $\mathsf{T}(f_i) = \tau_i$ for $i = 1, ..., r$.
Hence, taken a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\}$ and denoted by $I := I(\mathbf{X})$ the ideal of $\mathbf{X}$, the first step in order to find the factorized minimal Groebner basis $\mathcal{G} := \mathcal{G}(I(\mathbf{X}))$ of I is to compute the monomial basis $\mathsf{G}(I)$.
Clearly $\mathsf{G}(I)$ can be computed passing through the usual Groebner basis computation. Anyway, we want to do it in a pure combinatorial way, deriving $\mathsf{G}(I)$ from the Groebner escalier $\mathsf{N}(I) := \mathsf{N}(I(\mathbf{X}))$.
As explained in chapter 2, we can get $\mathsf{N}(I)$ directly from the points in $\mathbf{X}$ via the Cerlienco-Mureddu correspondence or the Felszeghy-B. Ráth-Rónyai Lex Game, them Gao-Rodrigues-Stroomer method or the Lederer's algorithm.
For the time being, we follow [79] and we only use Cerlienco-Mureddu Correspondence, but also the other methods work.
Moreover, in next chapter, we will study how to improve the Axis of Evil algorithm, exploiting a suitable method for computing the Groebner escalier.
At this stage, we can suppose as known:

- $\mathsf{N}(I)$, obtained via Cerlienco-Mureddu Correspondence;
- $\mathsf{G}(I)$, produced applying Lazard's algorithm to $\mathsf{N}(I)$.

The pseudocode of the algorithm is displayed in 5. For an implementation, see [103].

If the variables in $\mathcal{P} = \mathbf{k}[x_1, ..., x_n]$ are ordered as usual, namely $x_1 < x_2 < ... < x_n$, we know that the first generator τ_1 in $\mathsf{G}(I)$ is $\tau_1 = x_1^{d_{1,1}}$ for some $d_{1,1} \in \mathbb{N}$, since I is zerodimensional.
Computing the factors composing the polynomial $f_1 \in \mathcal{G}$ such that $\mathsf{T}(f_1) = \tau_1$ is particularly simple. Indeed, if $\tau_1 = x_1^{d_{1,1}} \in \mathsf{G}(I)$, then all the terms $1, x_1, ..., x_1^{d_{1,1}-1} \in \mathsf{N}(I)$.
As seen in chapters 1, 2, while discussing Moeller algorithm and the computational methods for the Groebner escalier, the condition $1, x_1, ..., x_1^{d_{1,1}-1} \in \mathsf{N}(I)$, means that the points in $\mathbf{X}$ have exactly $d_{1,1}$ different first coordinates.
Being an element of $\mathcal{G}$, f_1 has to vanish at all points of $\mathbf{X}$. Hence, if we compute the set

$$N_1(\tau_1) := \{x_1^i / \ i < d_{1,1}\} = \{\omega \in \mathcal{T}[1], \ \tau_1 > \omega x_2^{d_{1,2}} \cdots x_n^{d_{1,n}} \in \mathsf{N}(I)\},$$

being $d_{1,2} = ... = d_{1,n} = 0$, we get exactly $N_1(\tau_1) = \{1, x_1, ..., x_1^{d_{1,1}-1}\}$.
These terms correspond, by Cerlienco-Mureddu correspondence, to the first $d_{1,1}$ points with different first coordinates, say $A_1(\tau_1) = \{P_{\alpha_1}, ..., P_{\alpha_{d_{1,1}}}\}$.
For each $1 \leq j \leq d_{1,1}$, let a_j be the first coordinate of P_{a_j}. We let $B_1(\tau_1) = \{a_1, ..., a_{d_{1,1}}\}$ and

Algorithm 5 The Axis of Evil algorithm.

procedure AOE($\mathbf{X}, \mathsf{G}(I(\mathbf{X})) := \{\tau_1, ..., \tau_r\}) \to R$ ▷ R contains a factorized minimal Groebner basis of I.

Require: We denote $\tau_j = x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$ for $j = 1, ..., n$.

Ensure: Axis of Evil factorization.

$R = \emptyset$

for $i = 1$ to r **do**

$N_1(\tau_j) := \{x_1^i / \ i < d_{j,1}\} = \{\omega \in \mathcal{T}[1], \tau_j > \omega x_2^{d_{j,2}} \cdots x_n^{d_{j,n}} \in \mathsf{N}\}$

$A_1(\tau_j) := \{\Phi^{-1}(x_1^i x_2^{d_{j,2}} \cdots x_n^{d_{j,n}}) / \ i < d_{j,1}\} \subset \mathbf{X}$.

$B_1(\tau_j) := \pi_1(A_1(\tau_j)) \subset k$.

$\gamma_{1\tau_j} := \prod_{a \in B_1(\tau_j)} (x_1 - a)$.

for $m = 2$ to n **do**

$\zeta_{m\tau_j} := \prod_{\nu=1}^{m-1} \gamma_{\nu\tau_j}$.

$D_{m0} := \{P_i \in \mathbf{X} / \ \zeta_{m\tau_j}(P_i) \neq 0\}$.

if $|D_{m0}| = 0$ **then**

$R = [R, \zeta_{m\tau_j}]$.

break.

end if

$N_m(\tau_j) := \{\omega \in \mathcal{T}[m], \tau_j > \omega x_{m+1}^{d_{j,m+1}} \cdots x_n^{d_{j,n}} \in \mathsf{N}\}$.

for $\delta = 1$ to $d_{j,m}$ **do**

$A_{m\delta}(\tau_j) := \{\Phi^{-1}(v x_m^{d_{j,m}-\delta} x_{m+1}^{d_{j,m+1}} \cdots x_n^{d_{j,n}}) | v \in \mathcal{T}[m-1], v x_m^{d_{j,m}-\delta} \in N_m(\tau_j)\} \cap D_{m(\delta-1)}(\tau_j)$.

$E_{m\delta}(\tau_j) := \Phi(\pi_m(A_{m\delta}(\tau_j)))$.

$$\gamma_{m\delta\tau_j} := x_m + \sum_{\omega \in E_{m\delta}(\tau_j)} c(\gamma_{m\tau_j}, \omega)\omega,$$

such that $\gamma_{m\delta\tau_j}(P) = 0$, $\forall P \in A_{m\delta}(\tau_j)$.

$\xi_{m\delta} := \prod_{\nu=1}^{m-1} \gamma_{\nu\tau_j} \prod_{d=1}^{\delta} \gamma_{md\tau}$.

$D_{m\delta}(\tau_j) := \{P_i \in \mathbf{X} / \ \xi_{m\delta}(P_i) \neq 0\} \subseteq \mathbf{X}$

if $|D_{m\delta}(\tau_j)| = 0$ **then**

$R = [R, \xi_{m\delta}]$.

break.

end if

end for

$\gamma_{m\tau_j} := \prod_{\delta} \gamma_{m\delta\tau_j}$.

end for

end for

return R.

end procedure

we compute the polynomial

$$\gamma_{1\tau_1} := \prod_{j=1}^{d_{1,1}} (x_1 - a_j).$$

Since $\mathsf{T}(\gamma_{1\tau_1}) = \tau_1$ and $\gamma_{1\tau_1}$ vanishes over all $\mathbf{X}$, $f_1 = \gamma_{1\tau_1}$, so we have found the first element of $\mathcal{G}$.
Moreover, not only the factors composing f_1 but also f_1 itself is reduced, since $Supp(f_1) \setminus \{\tau_1\} \subseteq \{1, x_1, ..., x_1^{d_{1,1}-1}\} \subseteq \mathsf{N}(I)$. We point out that f_1 has been determined as the product of exactly $d_{1,1}$ factors.

Example 3.4.2. Let $\mathbf{X} = \{(1,0), (2,3), (4,6), (0,7), (5,2), (4,1), (2,6), (2,7), (0,6)\} \subset \mathbb{R}^2$.
We draw the unmixed tower structure we can get from $\mathbf{X}$ in order to have an overall view of the set.

Since we are dealing now with points in only 2 coordinates, Cerlienco-Mureddu algorithm turns out to be simplified. More precisely, we get the Groebner escalier by a tower reordering (2.2.8), so

$$\mathsf{N}(I) = \{1, x_1, x_1^2, x_1^3, x_1^4, x_2, x_1x_2, x_1^2x_2, x_2^2\}.$$

The monomial basis is $\mathsf{G}(I) = \{x_1^5, x_1^3x_2, x_1x_2^2, x_2^3\}$, so $\min_{Lex}(\mathsf{G}(I)) = x_1^5$. We get

$$N_1(\tau_1) := \{1, x_1, x_1^2, x_1^3, x_1^4\}$$

and its elements correspond to the points

$$A_1(\tau_1) = \{(2,3), (4,6), (0,7), (1,0), (5,2)\}.$$

The projection $\pi_1(A_1(\tau_1))$ is exactly the set containing the first coordinates, so it turns out to be

$$B_1(\tau_1) = \{2, 4, 0, 1, 5\}.$$

We obtain the polynomial (fulfilling the tasks of lines from 4 to 7 of algorithm 5)

$$f_1 = \gamma_{1\tau_1} = x_1(x_1-2)(x_1-4)(x_1-1)(x_1-5) = x_1^5 - 12x_1^4 + 49x_1^3 - 78x_1^2 + 40x_1,$$

clearly vanishing at all $\mathbf{X}$.
We know that f_1 belongs to the minimal Groebner basis of theorem 3.4.1, but it also belongs to the reduced Groebner basis, since $x_1, x_1^2, x_1^3, x_1^4 \in \mathsf{N}(I)$.
Actually, if we compute via Singular [30] the reduced Groebner basis of $I(\mathbf{X})$ we get

- $x_1^5 - 12x_1^4 + 49x_1^3 - 78x_1^2 + 40x_1$, that is exactly our f_1;
- $2x_1^3x_2 - 12x_1^2x_2 + 16x_1x_2 - x_1^4 + 7x_1^3 - 14x_1^2 + 8x_1$;
- $4x_1x_2^2 - 8x_2^2 + 6x_1^2x_2 - 64x_1x_2 + 104x_2 - 9x_1^4 + 107x_1^3 - 426x_1^2 + 664x_1 - 336$;
- $12x_2^3 - 192x_2^2 - 18x_1^2x_2 + 36x_1x_2 + 972x_2 - 149x_1^4 + 1583x_1^3 - 5218x_1^2 + 5296x_1 - 1512$.

We show now how to find f_j from $\tau_j = x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$, $j \leq r = |\mathsf{G}(I)|$. We refer to algorithm 5.
Similarly to what done for τ_1, we first study the first coordinates, namely we compute the set

$$N_1(\tau_j) := \{x_1^i / \ i < d_{j,1}\} = \{\omega \in \mathcal{T}[1],\, \tau_j > \omega x_2^{d_{j,2}} \cdots x_n^{d_{j,n}} \in \mathsf{N}(I)\}.$$

By Cerlienco-Mureddu correspondence, each term in $\mathsf{N}(I)$ is associated to a point of $\mathbf{X}$, so we can define

$$A_1(\tau_j) := \{\Phi^{-1}(x_1^i x_2^{d_{j,2}} \cdots x_n^{d_{j,n}}) / \ i < d_{j,1}\} \subset \mathbf{X}$$

and, if we project w.r.t the first coordinate, we get $B_1(\tau_j) := \pi_1(A_1(\tau_j)) \subset k$. The factors in x_1 are of the form $(x_1 - a)$ for $a \in B_1(\tau_j)$, so the partial factor in $x_1^{d_{j,1}}$ is

$$\gamma_{1\tau_j} := \prod_{a \in B_1(\tau_j)} (x_1 - a),$$

again following lines from 4 to 7 of algorithm 5.
We construct now the set

$$D_{20} := \{P_i \in \mathbf{X} / \ \gamma_{1\tau_j}(P_i) \neq 0\},$$

containing all the points in the given $\mathbf{X}$ such that $\gamma_{1\tau_j}$ do not vanish. If D_{20} is the empty set, then $f_j = \gamma_{1\tau_j}$. In this case, we do not have to deal with τ_j anymore[5] (we have executed what prescribed in lines 9-14).
Otherwise, we construct the set

$$N_2(\tau_j) := \{\omega \in \mathcal{T}[2],\, \tau_j > \omega x_3^{d_{j,3}} \cdots x_n^{d_{j,n}} \in \mathsf{N}(I)\},$$

containing the terms ω in the two variables x_1, x_2 such that $\tau_j > \omega x_3^{d_{j,3}} \cdots x_n^{d_{j,n}}$ in the Groebner escalier (line 15) and, for each δ from 1 to $d_{j,2}$, we compute the set of points where to interpolate, namely

$$A_{2\delta}(\tau_j) := \{\Phi^{-1}(v x_2^{d_{j,2}-\delta} x_3^{d_{j,3}} \cdots x_n^{d_{j,n}}) | v \in \mathcal{T}[1], v x_2^{d_{j,2}-\delta} \in N_2(\tau_j)\} \cap D_{2(\delta-1)}(\tau_j)$$

[5]It happens only for τ_1 since only one pure power of x_1 can occur in $\mathsf{G}(I)$, by the minimality of $\mathsf{G}(I)$.

and the set of terms appearing in the current factor, i.e. $E_{2\delta}(\tau_j) := \Phi(\pi_2(A_{2\delta}(\tau_j)))$.
With the above data, we perform the interpolation step and we finally get the factor

$$\gamma_{2,\delta\tau_j} := x_2 + \sum_{\omega \in E_{2\delta}(\tau_j)} c(\gamma_{2\tau_j}, \omega)\omega,$$

such that $\gamma_{2\delta\tau_j}(P) = 0$, $\forall P \in A_{2\delta}(\tau_j)$.
We compute then $D_{2\delta}(\tau_j) := \{P_i \in \mathbf{X} / \, \xi_{2\delta}(P_i) \neq 0\} \subseteq \mathbf{X}$, where $\xi_{2\delta}$ is the product of all the factors we have computed for τ_j. We stop if it is empty.
Repeating for each δ, we get all the factors with leading term x_2. The set $N_2(\tau_j)$ turns out to be partitioned w.r.t. the exponents of x_2[6] (and we have fulfilled the tasks of lines from 16 to 26).
At this point, we check whether the product of the current factors vanishes over all $\mathbf{X}$. If so, such a product is f_j, so we continue with another term in $\mathsf{G}(I)$. Otherwise, we repeat for $x_3, ..., x_n$, stopping the procedure for τ_j and storing f_j when we reach the last coordinate or when the product of the current factors vanish over all $\mathbf{X}$ (see line 8-14).
When f_j is stored, we perform in the same way with the other generators (line 3).
We point out that the polynomials $\gamma_{m\delta\tau_j}$ we get are only *linear in the leading terms*.
From now on we will call such a factorization (linear) *Axis of Evil factorization*.

Remark 3.4.3. By construction and essentially by Cerlienco-Mureddu correspondence and the consequent construction of the sets $E_{m\delta}(\tau_j)$, we get $\mathsf{T}(\gamma_{m\delta\tau_j}) = x_m$.

Even if algorithm 5 leans on Cerlienco-Mureddu correspondence, whose most important feature is iterativity on the points, it is *not iterative* on the elements of $\mathbf{X}$. Indeed all the Cerlienco-Mureddu biunivocal correspondence has to be known in order to proceed in the execution of the algorithm.

Remark 3.4.4. Let $\tau_j := x_1^{d_{j,1}} \cdots x_n^{d_{j,n}} \in \mathsf{G}(I)$. The required polynomial $f_j = \tau_j + tail(f_j) \in \mathcal{G}(I)$ has exactly $d_j = \sum_{i=1}^n d_{j,i}$ factors: $d_{j,1}$ with leading term x_1, $d_{j,2}$ with leading term x_2 and so on. As we can see in line 16 of algorithm, every variable x_i, $i = 1, ..., n$, appears only $d_{j,i}$ times in the execution of the algorithm.

Remark 3.4.5. The sets $N_m(\tau_j) := \{\omega \in \mathcal{T}[m], \tau_j > \omega x_{m+1}^{d_{j,m+1}} \cdots x_n^{d_{j,n}} \in \mathsf{N}(I)\}$ are constructed in order to find the points where one has to interpolate.
We point out that $N_m(\tau_j) \subseteq N_h(\tau_j)$ for $m \leq h$.
If $\omega \in N_m(\tau_j)$, $\omega \in \mathcal{T}[m]$ and $\tau_j > \omega x_{m+1}^{d_{m+1}} \cdots x_n^{d_n} \in \mathsf{N}(I)$. Since $m \leq h$, $\omega \in \mathcal{T}[h]$; as $\omega x_{h+1}^{d_{h+1}} \cdots x_n^{d_n} \mid \omega x_{m+1}^{d_{m+1}} \cdots x_n^{d_n}$ we have

[6] By computing the terms appearing in $A_{m\delta}(\tau_j)$.

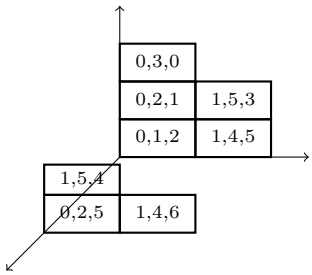
$\omega x_{h+1}^{d_{h+1}} \cdots x_n^{d_n} \in \mathsf{N}(I)$ and $\omega x_{h+1}^{d_{h+1}} \cdots x_n^{d_n} \leq x_{m+1}^{d_{m+1}} \cdots x_n^{d_n} < \tau_j$.
Since for each term $\mu \in \mathsf{N}(I)$ such that $\mu > \tau_j$, Cerlienco-Mureddu provides a point $P_{\mu'}$ such that $\exists k \in \{1, ..., n\}$ $P_\mu, P_{\mu'}$ have the first k coordinates and $\mu' < \mu$, in order to obtain polynomials vanishing on all the points of $\mathbf{X}$ it is not necessary to interpolate in the whole $\Phi^{-1}(\mathsf{N})$ as it suffices to consider only those corresponding to $\mu \in \mathsf{N}(I)$ with $\mu < \tau_j$.

Example 3.4.6. Consider the set

$$\mathbf{X} = \{(0,1,2), (1,4,5), (0,2,1), (1,5,3), (0,3,0), (0,2,5), (1,4,6), (1,5,4)\}$$

and denote, as usual, $I := I(\mathbf{X})$.
As shown in the (mixed) tower structure below, the Groebner escalier of its associated ideal is

$$\mathsf{N}(I) = \{1, x_1, x_2, x_1x_2, x_2^2, x_3, x_1x_3, x_2x_3\}.$$



The monomial basis is then $\mathsf{G}(I) = \{x_1^2, x_1x_2^2, x_2^3, x_1x_2x_3, x_2^2x_3, x_3^2\}$.
We focus on $\tau_2 = x_1x_2^2$ and we observe that $x_2x_3 \in \mathsf{N}(I)$ is greater than τ_2 w.r.t. the lexicographical order induced by $x_1 < x_2 < x_3$.
With the notation due to Cerlienco-Mureddu we can say that $\Phi^{-1}(x_2x_3) = (1,5,4)$, and we can notice that:

- the factor $x_2 - 5$ produced in order to make f_2 vanish on the point $(1,5,3)$ makes also f_2 vanish on the point $(1,5,4)$, since $\pi_2(1,5,3) = (1,5) = \pi_2(1,5,4)$;
- we have $(1,5,3) = \Phi^{-1}(x_1x_2)$ and $x_1x_2 < \tau_2$.

For completeness' sake, we report here the whole Axis of Evil factorization of I, computed using Singular:

x_1^2: $f_1 = x_1(x_1 - 1)$;

$x_1x_2^2$: $f_2 = x_1(x_2 - 5)(x_2 - 4)$;

x_2^3: $f_3 = (x_2 - 3)(x_2 - 3x_1 - 2)(x_2 - 3x_1 - 1)$;

$x_1x_2x_3$: $f_4 = (x_1 - 1)(x_2 - 2)(x_3 + x_2 - 3)$;

$x_2^2x_3$: $f_5 = (x_2 - 5)(x_2 - 2x_1 - 2)(x_3 + x_2 - 3)$

x_3^2: $f_6 = (x_3 + 2x_2 - 5x_1 - 9)(x_3 + x_1x_2 + x_2 - 10x_1 - 3)$.

Remark 3.4.7. The terms mentioned in remark 3.4.5, smaller than the current τ_j, are found "releasing" all the variables one by one.
Imagining the terms in $\mathcal{T}$ as points in $\mathbb{N}^n$ (each term is identified with the n-tuple of its exponents, see chapter 1) we can think of our releasing as an *increment by one of the 'directions' where we can move*.
At each step we count out all the points in which the polynomial already vanishes and we stop the computation when the current factorized polynomial vanishes on the whole $\mathbf{X}$.

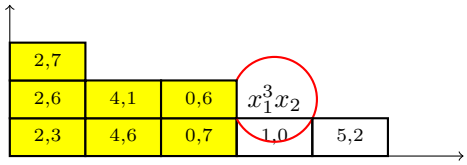
Example 3.4.8. Consider again the set

$$\mathbf{X} = \{(1,0),(2,3),(4,6),(0,7),(5,2),(4,1),(2,6),(2,7),(0,6)\} \subset \mathbb{R}^2$$

of example 3.4.2.
We point out that in the first step for τ_j, while computing $N_1(\tau_j)$ and $A_1(\tau_j)$, we release only x_1 and we list the terms of the form $x_1^i x_2^{d_{j,2}} \cdots x_n^{d_{j,n}}$, so the ones with the same exponents as τ_j in $x_2, ..., x_n$, which correspond to points lying in a *higher tower* than the one over which τ_j lies.
We have

$$\mathsf{N}(I) = \{1, x_1, x_1^2, x_1^3, x_1^4, x_2, x_1x_2, x_1^2x_2, x_2^2\},$$

and the monomial basis is $\mathsf{G}(I) = \{x_1^5, x_1^3x_2, x_1x_2^2, x_2^3\}$.
For example, focus on $\tau_2 = x_1^3x_2$. For this term we have $N_1(\tau_2) = \{1, x_1, x_1^2\}$ and, consequently, $A_1(\tau_2) = \{(2,6),(4,1),(0,6)\}$. As shown in the (unmixed) tower structure below, the terms belong to towers higher than the one over which τ_2 lies:



Remark 3.4.9. For each $\delta \in \{0, ..., d_{j,m}\}$ and for each $\tau_j \in \mathsf{G}(I(\mathbf{X}))$, $\tau_j \neq \tau_1$, define the sets

$$S_{m\delta}(\tau_j) := \{v x_m^{d_{j,m}-\delta} \in N_m(\tau_j),\ v \in \mathcal{T}[m-1]\} \subset N_m(\tau_j).$$

Notice that, for $\delta_1, \delta_2 \in \{0, ..., d_{j,m}\}$, $\delta_1 \neq \delta_2$, we get $S_{m\delta_1}(\tau_j) \cap S_{m\delta_2}(\tau_j) = \emptyset$ and that $N_m(\tau_j) = \bigcup_{\delta=0}^{d_{j,m}} S_{m\delta}(\tau_j)$, so the subsets $S_{m\delta}(\tau_j)$ which are nonempty form a partition of

$N_m(\tau_j)$.
Even if in Algorithm 5 there is no need to define explicitly the subsets $S_{m\delta}(\tau_j)$, those for $\delta \in \{1, ..., d_{j,m}\}$ are essentially used in the construction of the sets $A_{m\delta}(\tau_j)$, $\delta \in \{1, ..., d_{j,m}\}$ (see line 17). This means that the subsets $S_{m\delta}(\tau_j)$ come into play in the choice of the points where to interpolate while constructing of the current factor.
Notice that

$$S_{m0}(\tau_j) = \{v x_m^{d_{j,m}} \in N_m(\tau_j),\ v \in \mathcal{T}[m-1]\} \subset N_m(\tau_j).$$

is not used in the construction (in line 16 we consider $\delta = 1, ..., d_{j,m}$), even if by any chance $S_{m0}(\tau_j) \neq \emptyset$. Actually, it holds $S_{m0}(\tau_j) \subseteq N_{m-1}(\tau_j)$, so each $\sigma \in S_{m0}(\tau_j)$ has already been considered: the current factorized polynomial already vanishes in $\Phi^{-1}(\sigma x_{m+1}^{d_{j,m+1}} \cdots x_n^{d_{j,n}})$.

Remark 3.4.10. The steps made by the algorithm on each τ_j are totally independent both on those made and on those to be made on a term τ_k (it is indifferent whether $j \gtrless k$) belonging to $\mathsf{G}(I)$, so we will obtain the same factorizations even if we launch the computation on a list of unordered terms.
Clearly, the result of our computation is not the reduced Groebner basis of the given ideal, it is only one of the minimal Groebner bases but we can obtain the reduced Groebner basis via simple reduction.

Example 3.4.11. Consider again the set

$$\mathbf{X} = \{(0,1,2), (1,4,5), (0,2,1), (1,5,3), (0,3,0), (0,2,5), (1,4,6), (1,5,4)\}$$

of example 3.4.6 and denote, $I := I(\mathbf{X})$.
We already know the Axis of Evil factorization by example 3.4.6, but now we reduce all the polynomials.
The underlined terms represent the ones we have to reduce.

x_1^2: $f_1 = x_1^2 - x_1$ is already reduced.

$x_1x_2^2$: $f_2 = x_1x_2^2 - 9x_1x_2 + 20x_1$ is again reduced, so there is nothing to do.

x_2^3: $f_3 = x_2^3 - 6\underline{x_1x_2^2} - 6x_2^2 + 9\underline{x_1^2x_2} + 27x_1x_2 + 11x_2 - 27\underline{x_1^2} - 27x_1 - 6$ is not reduced. We have to reduce it using f_1, f_2, obtaining $f_3' = x_2^3 - 6x_2^2 - 18x_1x_2 + 11x_2 + 66x_1 - 6$.

$x_1x_2x_3$: $f_4 = x_1x_2x_3 - x_2x_3 - 2x_1x_3 + 2x_3 + \underline{x_1x_2^2} - x_2^2 - 5x_1x_2 + 5x_2 + 6x_1 - 6$ has to be reduced using f_2. One gets $f_4' = x_1x_2x_3 - x_2x_3 - 2x_1x_3 + 2x_3 - x_2^2 + 4x_1x_2 + 5x_2 - 14x_1 - 6$.

$x_2^2x_3$: $f_5 = x_2^2x_3 - 2\underline{x_1x_2x_3} - 7x_2x_3 + 10x_1x_3 + 10x_3 + \underline{x_2^3} - 2\underline{x_1x_2^2} - 10x_2^2 + 16x_1x_2 + 31x_2 - 30x_1 - 30$ is not reduced. We have to preform Buchberger reduction on it using f_2, f_3 and we get $f_5' = x_2^2x_3 - 9x_2x_3 + 6x_1x_3 + 14x_3 - 6x_2^2 + 24x_1x_2 + 30x_2 - 84x_1 - 36$.

x_3^2: $f_6 = x_3^2 + \underline{x_1x_2x_3} + 3x_2x_3 - 15x_1x_3 - 12x_3 + 2\underline{x_1x_2^2} + 2x_2^2 - 5\underline{x_1^2x_2} - 34x_1x_2 - 15x_2 + 50\underline{x_1^2} + 105x_1 + 27$ is not reduced. If we use f_1, f_2, f_4 on it, we obtain $f_6' = x_3^2 + 4x_2x_3 - 13x_1x_3 - 14x_3 + 3x_2^2 - 25x_1x_2 - 20x_2 + 129x_1 + 33$.

The reduced Groebner basis turns then out to be

$$\begin{aligned}\mathcal{G}' = \{&x_1^2 - x_1, x_1x_2^2 - 9x_1x_2 + 20x_1, x_2^3 - 6x_2^2 - 18x_1x_2 + 11x_2 + 66x_1 - 6,\\ &x_1x_2x_3 - x_2x_3 - 2x_1x_3 + 2x_3 - x_2^2 + 4x_1x_2 + 5x_2 - 14x_1 - 6,\\ &x_2^2x_3 - 9x_2x_3 + 6x_1x_3 + 14x_3 - 6x_2^2 + 24x_1x_2 + 30x_2 - 84x_1 - 36,\\ &x_3^2 + 4x_2x_3 - 13x_1x_3 - 14x_3 + 3x_2^2 - 25x_1x_2 - 20x_2 + 129x_1 + 33\}\end{aligned}$$

Remark 3.4.12 ([73])**.** Notice that the sets $E_{m\delta}(\tau_j)$ and the interpolating polynomials $\gamma_{m\delta\tau_j}$ of algorithm 5 can be obtained via Moeller algorithm and projection through π_m of the points found $A_{m\delta}(\tau_j)$, as well as via Cerlienco-Mureddu Correspondence and other interpolation methods.

Remark 3.4.13. Fix a term $\tau_j \in \mathsf{G}(I)$. If some $P = (a_1, ..., a_n) \in \mathbf{X}$ belongs to $A_{m\delta}(\tau_j)$, $2 \leq m \leq n, 1 \leq \delta \leq d_{j,m}$, then the linear factor vanishing in P, namely $\gamma_{m\delta\tau_j}$, is constructed involving only the first m coordinates of P, i.e. $a_1, ..., a_m$.

Remark 3.4.14. Although the minimal Groebner basis we get by the Axis of Evil algorithm is not reduced, we can point out that the linear factors $\gamma_{m\delta\tau_j}$ we get are reduced in the sense that $Supp(\gamma_{m\delta\tau_j}) \setminus \{x_m\} \subseteq \{\tau \in \mathsf{N}(I) \,|\, \tau < x_m\}$ by the construction of $E_{m\delta}(\tau_j)$.

Example 3.4.15. If we consider the set $\mathbf{X} = \{(0,0), (1,2), (0,2), (3,4), (0,6)\}$, the minimal Groebner basis produced by the Axis of Evil algorithm is

$$\mathcal{G} = \{x^3 - 4x^2 + 3x, xy - x^2 - x, y^3 - \frac{4}{3}xy^2 - 8y^2 + \frac{32}{3}xy + 12y - 16x\},$$

and the linear factors identifying $\mathcal{G}$ are

a. x;

b. $x - 1$;

c. $x - 3$;

d. $y - x - 1$;

e. $y - 6$;

f. $y - 2$;

g. $y - \frac{4}{3}x$.

Factors a, b, c, e, f are of the form $x - l, y - h$, with l, h constants, so their support is formed by the leading terms x or y and by $1 \in \mathsf{N}$. Factors d and g satisfy again the property of 3.4.14, since

- $Supp(y - x - 1) \setminus \{y\} = \{1, x\} \subset \mathsf{N}(I)$ and $1 < x < y$;
- $Supp(y - \frac{4}{3}x) \setminus \{y\} = \{x\} \subset \mathsf{N}(I)$ and $x < y$.

Developing an algorithm one has to face the problems of *termination* and *correctness*.
As for our algorithm, termination is guaranteed since it essentially operates with the following three loops:

– a loop on the elements of $\mathsf{G}(I)$ (line 3);

– a loop on the variables of the polynomial ring (line 8);

– for each variable appearing in a term $\tau_j \in \mathsf{G}(I)$, a loop on its exponent (line 16).

The first loop is clearly finite by Dickson's Lemma (c.f. [79]), while the second is finite since the polynomial ring has a finite number of variables.
As regards the third one, it is trivially finite since the exponents are natural numbers.
The algorithm could go to infinity if it was $|\mathsf{N}(I)| = \infty$, but this is not the case for our zerodimensional radical ideal I. Finally, it relies on Cerlienco-Mureddu algorithm and Moeller algorithm so also the computation of the set $A_{m\delta}(\tau_j)$ and the interpolation step terminate.
Let us study the correctness of the algorithm.

Lemma 3.4.16. The obtained factorized polynomials vanish on each point of $\mathbf{X}$.

Proof: Consider the polynomial γ_τ associated to the term $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{G}(I)$.
We prove that it vanishes on $P_\mu \in \mathbf{X}$, corresponding, via Cerlienco-Mureddu , to the term $\mu = x_1^{\beta_1} \cdots x_n^{\beta_n} \in \mathsf{N}(I)$.
Since $\tau \in \mathsf{G}(I)$ and $\mu \in \mathsf{N}(I)$, $\tau \neq \mu$. Therefore, there are only two possibilities:

1. $\mu <_{Lex} \tau$. By the definition of Lex, $\exists i \in \{1, ..., n\}$ such that $\alpha_i > \beta_i$ and $\alpha_j = \beta_j$ for each $i + 1 \leq j \leq n$, so $\beta_i = \alpha_i - \delta$, for some $\delta > 0$. We set $\omega := x_1^{\beta_1} \cdots x_i^{\beta_i}$. By hypothesis, $\mu \in \mathsf{N}(I)$ and $\mu = \omega x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} < \tau$, so $\omega \in N_i(\tau)$.
   As $P_\mu = \Phi^{-1}(\mu) = \Phi^{-1}(x_1^{\beta_1} \cdots x_{i-1}^{\beta_{i-1}} x_i^{\alpha_i - \delta} x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n})$, either $P_\mu \in D_{i(\delta-1)}(\tau)$ (thus γ_τ vanishes in P_μ), or $P_\mu \in A_{i\delta}(\tau)$ but, in this case, by the interpolation step (lines 18-19), γ_τ vanishes in P_μ.

2. $\mu >_{Lex} \tau$. This time $\exists i \in \{1, ..., n\}$ such that $\beta_i > \alpha_i$, $\beta_j = \alpha_j$ for each $j \in \{i+1, ..., n\}$. By Cerlienco-Mureddu correspondence, $\exists \mu' := x_1^{\beta'_1} \cdots x_n^{\beta'_n} \in \mathsf{N}(I)$ such that:

   a. $\Phi^{-1}(\mu') = P_{\mu'}$ with $\pi_{i-1}(P_\mu) = \pi_{i-1}(P_{\mu'})$;

   b. $\beta'_h = \alpha_h, \forall h \in \{i, i+1, ..., n\}$.

   If $\mu' < \tau$, then $\mu' \in N_{i-1}(\tau)$ so, as in 1, γ_τ vanishes in $P_{\mu'}$ and the linear factor making γ_τ vanish in $P_{\mu'}$ is computed involving at most the first $i-1$ coordinates of P_μ (c.f. remark 3.4.13), so that γ_τ turns out to vanish also in P_μ.
   If $\mu' > \tau$, we can repeat with μ' instead of μ and conclude by induction.

$\diamond$

Corollary 3.4.17. The ideal generated by our polynomials is exactly $I(\mathbf{X})$.

Proof: By lemma, 3.4.16, the polynomials vanish on all the points of the set $\mathbf{X}$ and the equality comes out by multiplicity reasons. $\diamond$

Algorithm 5 and lemma 3.4.16 constitute a constructive proof of the Axis of Evil Theorem 3.4.1.

Remark 3.4.18. The polynomials $f_1, ..., f_r$ of theorem 3.4.1 form a minimal Groebner basis because:

- they vanish on all the points of $\mathbf{X}$ (lemma 3.4.16);
- their heads $\mathsf{T}(f_1) = \tau_1, ..., \mathsf{T}(f_r) = \tau_r$ form exactly $\mathsf{G}(I(\mathbf{X}))$.

Remark 3.4.19. We point out that:

- if $\tau_j = x_1^{d_{j,1}} \cdots x_n^{d_{j,n}} \in \mathsf{G}(I)$, the polynomials we are looking for have to contain *exactly* $\sum_{i=1}^n d_i$ factors. It is impossible that a partial product vanishes on the whole $\mathbf{X}$. In fact, if so, there would be a polynomial $f \in I$ such that $\mathsf{T}(f) \notin (\mathsf{G}(I))$.
- if we obtain a factorized polynomial f such that its leading term $\mathsf{T}(f)$ belongs to the minimal basis $\mathsf{G}(I)$, then f vanishes over all $\mathbf{X}$, because of 3.4.16.

This implies that the termination criteria for algorithm 5 are correct.

Remark 3.4.20. Cerlienco-Mureddu Correspondence is performed on an ordered set of points and this ordering influences the biunivocal correspondence we get. For example,

if we consider first the set $\underline{\mathbf{X}_1} = (P_1 = (1,0), P_2 = (1,1))$ we obtain $\Phi(P_1) = 1,\ \Phi(P_2) = x_2$, whereas if we have $\underline{\mathbf{X}_2} = (P_2 = (1,1), P_1 = (1,0))$, we obtain $\Phi(P_2) = 1,\ \Phi(P_1) = x_2$.
The Axis of Evil algorithm works correctly for each biunivocal correspondence we can get by ordered sets of points (so also with the biunivocal correspondences we can recover from another method for the Groebner escalier).

It is well known that Cerlienco-Mureddu correspondence allows to compute the Groebner escalier of zerodimensional ideals, even if they are not radical. Unfortunately, in general, it is not possible to produce an Axis of Evil factorization in case of multiplicity.
We display here a meaningful counterexample, due to M.G. Marinari and T. Mora.

Example 3.4.21 ([70, 79])**.** Consider the following ideal, given with its primary decomposition:
$J := (x_1^2, x_2 + x_1, x_3) \cap (x_1^2, x_2 - x_1, x_3 - 1) =$
$= (x_1^2, x_1x_2, x_2^2, x_1x_3 - \frac{1}{2}x_1 - \frac{1}{2}x_2, x_2x_3 - \frac{1}{2}x_1 - \frac{1}{2}x_2, x_3^2 - x_3) \lhd \mathbb{C}[x_1, x_2, x_3]$.
Denote by $f_1, ..., f_6$ the generators.
J is 0-dimensional being $x_1^2, x_2^2, x_3^2 \in \mathsf{T}(J)$ (see [79]), but it is not radical as

$$\sqrt{J} = (x_2, x_3^2 - x_3, x_1).$$

For such an ideal the Axis of Evil does not hold.
Consider the polynomial $f_4 = x_1x_3 - \frac{1}{2}x_1 - \frac{1}{2}x_2$.
By the Axis of Evil theorem (3.4.1), its factorization should be of the form:

$$(x_1 + ...)(x_3 + ...)$$

and we should have

$$x_1x_3 - \frac{1}{2}x_1 - \frac{1}{2}x_2 + Px_1^2 + Qx_1x_2 + Rx_2^2,\ P, Q, R \in \mathbb{C}[x_1, x_2, x_3],$$

we can only reduce deleting the multiples of x_1^2, x_1x_2, x_2^2, in order to obtain f_4 so we must have $-\frac{1}{2}x_2$ in it. We can not obtain it through reductions, so the only chance is to have a product of the form

$$k * hx_2,$$

with h, k constants such that $hk = -\frac{1}{2}$, in particular both different from 0.
A priori, there are two possibilities:
- $(x_1 + k)(x_3 + hx_2 + ...)$;
- $(x_1 + hx_2 + ...)(x_3 + k + ...)$.

The second one is impossible: the polynomial having x_1 as head can not contain variables greater than x_1, so we consider only:

$$(x_1 + k + ...)(x_3 + hx_2 + ...) \text{ obtaining } x_1x_3 + hx_1x_2 + kx_3 - \frac{1}{2}x_2 + ...$$

We can delete the term x_1x_2 but kx_3 can not be reduced.

The Axis of Evil Theorem can be generalized in case of Cerlienco-Mureddu ideals (see [79] for more details).

3.5 Consequences of the Axis of Evil Theorem.

We enumerate here some theorems which can be viewed as "corollaries" of the Axis of Evil Theorem (see , for example, [2]), quoting their general statements. Clearly they can only be deduced by 3.4.1 under our hypotheses.

We start with *Lazard Structural Theorem* 3.3.15, concerning minimal lexicographical Groebner basis of an ideal $I \triangleleft \mathbf{k}[x_1, x_2]$. The original proof, viewing $\mathbf{k}[x_1, x_2]$ as $\mathbf{k}[x_1][x_2]$, strongly uses that $\mathbf{k}[x_1]$ is a Principal Ideal Domain (PID). Norton-Sălăgean [81] reformulated it for $R[x]$ with R any PIR[7].
Next result is the one by Norton-Sălăgean.

Theorem 3.5.1 (Norton-Sălăgean)**.** With the notation of theorem 3.3.15, each

$$H_{i+1} \in (f_j, j < i) : r_i.$$

In fact, we have $r_i = \prod_{m=1}^{n-1} \prod_{\delta=1}^{d_m} \gamma_{m\delta t_i}$ and $H_i = \prod_{\delta=1}^{d_n} \gamma_{n\delta t_i}$.

The next result is *Kalkbrener theorem* ([60], [79]), which is a stronger characterization of the lexicographical ordering.
For each subset $\mathcal{G} \subset \mathbf{k}[x_1, ..., x_n]$, $i \in \{1, ..., n\}$, $\forall \delta \in \mathbb{N}$ set

$\mathcal{G}_{i,\delta} = \{p \in \mathcal{G} \mid p \in \mathbf{k}[x_1, ..., x_i],\ deg_i(p) \leq \delta\}$ and $Lp_{i,\delta}(\mathcal{G}) = \{Lp(p),\ p \in \mathcal{G}_{i,\delta}\} \subseteq \mathbf{k}[x_1, ..., x_{i-1}]$.

Theorem 3.5.2 (Kalkbrener)**.** With the previous notation, let $I \triangleleft \mathbf{k}[x_1, ..., x_n]$ be an ideal. Then the following are equivalent:

- $\mathcal{G}$ is a Groebner basis of I w.r.t, the lexicographical order $<$ induced by $x_1 < ... < x_n$;
- $Lp_{i,\delta}(\mathcal{G})$ is a Groebner basis of $Lp_{i,\delta}(I)$, $i = 1, ..., n$, $\forall \delta \in \mathbb{N}$.

[7] Principal ideal ring.

Finally we mention *Gianni-Kalkbrener theorem*, whose situation is a bit more complicated (see [59], [44], [79]).

Theorem 3.5.3 (Gianni-Kalkbrener)**.** Consider the lex order induced by $x_1 < ... < x_n$ and a zerodimensional ideal $I \lhd \mathbf{k}[x_1, ..., x_n]$ with Groebner basis $\mathcal{G}_{,}$ whose elements are increasingly ordered w.r.t. lex on the leading terms, and $\mathcal{G}_d = \mathcal{G} \cap \mathbf{k}[x_1, ..., x_d]$. For $\alpha = (b_1, ..., b_d) \in V(I_d)$ define the projection map

$$\Phi_\alpha : \mathbf{k}[x_1, ..., x_n] \to \mathbf{k}[x_{d+1}, ..., x_n] \text{ s.t. } f(x_1, ..., x_n) \mapsto f(b_1, ..., b_d, x_{d+1}, ..., x_n).$$

Let σ be the minimal value s.t. $\Phi_\alpha(Lp(g_\sigma)) \neq 0$ and j, δ the values s.t.

$$g_\sigma = Lp(g_\sigma)x_j^{\delta+1} + ... \in \mathbf{k}[x_1, ..., x_j] \setminus \mathbf{k}[x_1, ..., x_{j-1}].$$

Then

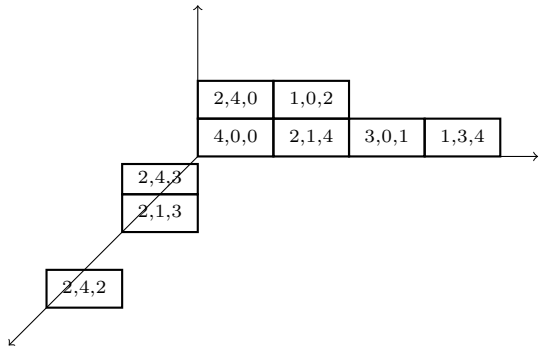
1. $j = d + 1$
2. $\forall g \in \mathcal{G}_d$, $\Phi_\alpha(g) = 0$;
3. $\forall g \in \mathcal{G}_{d+1,\delta}$, $\Phi_\alpha(g) = 0$;
4. $\Phi_\alpha(g_\sigma) = gcd(\Phi_\alpha(g),\ g \in G_{d+1}) \in \mathbf{k}[x_{d+1}]$;
5. $\forall b \in k$, $(b_1, ..., b_d, b) \in V(I_{d+1}) \Leftrightarrow \Phi_\alpha(g_\sigma)(b) = 0$.

Clearly $(1\text{-}3)$ are essentially a corollary of theorem 3.5.1; on the other side, $(4\text{-}5)$ apparently cannot be deduced from the Axis of Evil Theorem.

3.6 The Axis of Evil in pratice: a detailed example.

In this paragraph, we simulate in detail the Axis of Evil algorithm, giving a precise example of its main features.
We will examine the redistribution performed on the given points using their mixed tower structure.
Consider the set
$\mathbf{X} := \{(4,0,0), (2,1,4), (2,4,0), (3,0,1), (2,1,3), (1,3,4), (2,4,3), (2,4,2), (1,0,2)\}$.
First of all, we apply Cerlienco-Mureddu algorithm on $\mathbf{X}$.
$P_1 := (4,0,0)$: is a single point, so $\Phi(\{(4,0,0)\}) = (0,0,0)$
$P_2 := (2,1,4) : s = 1, m = 1, (1,0,0)$
$P_3 := (2,4,0) : s = 2, m = 2, (0,1,0)$
$P_4 := (3,0,1) : s = 1, m = 1, (2,0,0)$

$P_5 := (2,1,3) : s = 3, m = 2, (0,0,1)$
$P_6 := (1,3,4) : s = 1, m = 4, (3,0,0)$
$P_7 := (2,4,3) : s = 3, m = 3, \mathcal{W} = \{(2,1,3),(2,4,3)\}, t_7 = (0,1,1)$
$P_8 := (2,4,2) : s = 3, m = 7, (0,0,2)$
$P_9 := (1,0,2) : s = 2, m = 6, \mathcal{W} = \{(2,4,0),(1,0,2)\}, t_9 = (1,1,0).$
Then $\mathsf{N} := \{1, x_1, x_2, x_1^2, x_3, x_1^3, x_2x_3, x_3^2, x_1x_2\}$.
We display here the tower structure individuated by Cerlienco-Mureddu correspondence between $\mathbf{X}$ and N.



We apply now Lazard algorithm in order to get the monomial basis:

$1:$ we get $L = [x_1, x_2, x_3]$ and $\mathsf{G}_1 = \{x_1, x_2, x_3\}$;

$x_1:$ removing x_1 and computing the products, we have $L = \{x_1^2, x_2, x_1x_2, x_3, x_1x_3\}$, so $\mathsf{G}_2 = \{x_1^2, x_2, x_3\}$;

$x_2:$ $L = \{x_1^2, \underbrace{x_1x_2}_{\text{2times}}, x_2^2, x_3, x_1x_3, x_2x_3\}$, so $\mathsf{G}_3 = \{x_1^2, x_1x_2, x_2^2, x_3\}$;

$x_1^2:$ $L = \{x_1^3, \underbrace{x_1x_2}_{\text{2times}}, x_1^2x_2, x_2^2, x_3, x_1x_3, x_1^2x_3, x_2x_3\}$, so $\mathsf{G}_4 = \{x_1^3, x_1x_2, x_2^2, x_3\}$;

$x_3:$ $L = \{x_1^3, \underbrace{x_1x_2}_{\text{2times}}, x_1^2x_2, x_2^2, \underbrace{x_1x_3}_{\text{2times}}, x_1^2x_3, \underbrace{x_2x_3}_{\text{2times}}, x_3^2\}$, so
$\mathsf{G}_5 = \{x_1^3, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2\}$;

$x_1^3:$ $L = \{x_1^4, \underbrace{x_1x_2}_{\text{2times}}, x_1^2x_2, x_1^3x_2, x_2^2, \underbrace{x_1x_3}_{\text{2times}}, x_1^2x_3x_1^3x_3, \underbrace{x_2x_3}_{\text{2times}}, x_3^2\}$, so
$\mathsf{G}_6 = \{x_1^4, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2\}$;

$x_2x_3:$ $L = \{x_1^4, \underbrace{x_1x_2}_{\text{2times}}, x_1^2x_2, x_1^3x_2, x_2^2, \underbrace{x_1x_3}_{\text{2times}}, x_1^2x_3x_1^3x_3, x_1x_2x_3, x_2^2x_3, x_3^2, x_2x_3^2\}$, so
$\mathsf{G}_7 = \{x_1^4, x_1x_2, x_2^2, x_1x_3, x_3^2\}$;

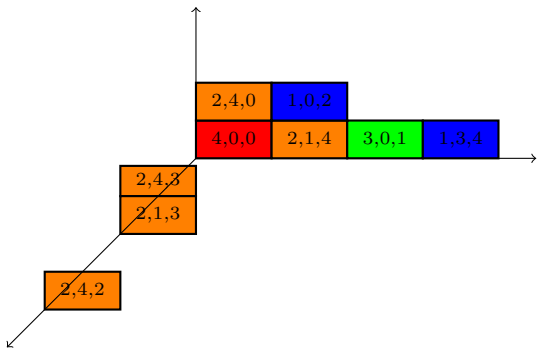
$x_3^2:\ L = \{x_1^4, \underbrace{x_1x_2}_{\text{2times}}, x_1^2x_2, x_1^3x_2, x_2^2, \underbrace{x_2x_3}_{\text{2times}}, x_1^2x_3x_1^3x_3, x_1x_2x_3, x_2^2x_3, x_1x_3^2, \underbrace{x_2x_3^2}_{\text{2times}}, x_3^3\}$, so
$\mathsf{G}_8 = \{x_1^4, x_1x_2, x_2^2, x_1x_3, x_2x_3^2, x_3^3\}$;

$x_1x_2:\ L = \{x_1^4, \underbrace{x_1^2x_2}_{\text{2times}}, x_1^3x_2, x_2^2, x_1x_2^2, \underbrace{x_1x_3}_{\text{2times}}, x_1^2x_3x_1^3x_3, \underbrace{x_1x_2x_3}_{\text{2times}}, x_2^2x_3, x_1x_3^2, \underbrace{x_2x_3^2}_{\text{2times}}, x_3^3\}$,
so $\mathsf{G}_9 = \{x_1^4, x_1^2x_2, x_2^2, x_1x_3, x_2x_3^2, x_3^3\}$.

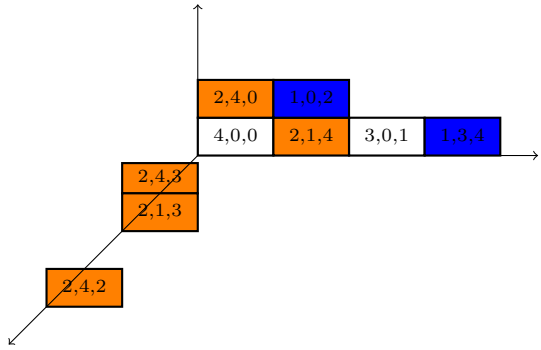
Then we obtain

$$\mathsf{G} = \{x_1^4, x_1^2x_2, x_2^2, x_1x_3, x_2x_3^2, x_3^3\}$$

The terms in G are exactly the input for the Axis of Evil algorithm and they are already ordered w.r.t. our ordering. We denote them by τ_i for $i = 1, ..., 6$.
Starting with $\tau_1 = x_1^4$ we obtain:
$N_1(\tau_1) = \{1, x_1, x_1^2, x_1^3\}$,
$A_1(\tau_1) = \{(4,0,0), (2,1,4), (3,0,1), (1,3,4)\}$: these are the corresponding points via Cerlienco-Mureddu;
$B_1(\tau_1) = \{4, 2, 3, 1\}$
$\gamma_{1\tau_1} = (x_1 - 4)(x_1 - 2)(x_1 - 3)(x_1 - 1)$: all the linear factors are only depending from x_1 and they are computed in the same time.
We highlight in the picture the points making $\gamma_{1\tau_1}$ vanish and we distinguish them, using colours, w.r.t. the linear factor vanishing on them (i.e. w.r.t. their first coordinates).



$m = 2$: $\zeta_{2\tau_1} = \gamma_{1\tau_1}$. Since, as we can also see in the picture above, $D_{20}(\tau) = \emptyset$, we stop here obtaining, as first result, a polynomial $f_1 := \zeta_{2\tau_1} = \gamma_{1\tau}$ whose leading term is $\tau_1 \in \mathsf{G}$, while the lower monomials belong to N. By construction, $f_1 \in I(\mathbf{X})$, since it vanishes in every point of $\mathbf{X}$. It is then an element of our minimal Groebner basis.
For $\tau_2 = x_1^2x_2$ we get: $N_1(\tau_2) = \{1, x_1\}$, $A_1(\tau_2) = \{(2,4,0), (1,0,2)\}$, $B_1(\tau_2) = \{2, 1\}$, $\gamma_{1\tau_2} = (x_1 - 2)(x_1 - 1)$



Passing to $m = 2$ we have:

$\zeta_{m\tau_2} = \gamma_{1\tau_2}$

$D_{20}(\tau_2) = \{(4,0,0),(3,0,1)\}$ (the two non-colored points in the picture above).

We cannot stop here, since we got a polynomial *not vanishing at all the points*. Moreover, we point out that its head is different from $\tau_2 \in \mathsf{G}$.

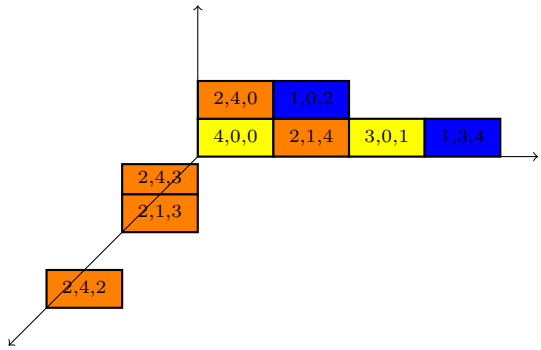
$N_2(\tau_2) = \{1, x_1, x_1^2, x_1^3, x_2, x_1x_2\}$; doing so, we find all the terms of the previous step and some new ones. We start the loop on δ:

$\delta = 1$: $A_{21}(\tau_2) = \{(4,0,0),(3,0,1)\} = D_{20}$

The terms $vx_m^{d_m-\delta}$ are $1, x_1, x_1^2, x_1^3$, corresponding to the points P_1, P_2, P_4, P_6.

Since the polynomial already vanishes on P_2, P_6, we consider only the other two points.

$E_{21}(\tau_2) = \{1, x_1\}$, $\gamma_{21\tau_2} = x_2$; $\xi_{21} = \gamma_{1\tau_2}\gamma_{21\tau_2} = (x_1-2)(x_1-1)x_2$; $D_{21}(\tau_2) = \emptyset$:



Remark that $\gamma_{2\tau_2}$ is actually $\gamma_{21\tau_2}$.

$\tau_3 = x_2^2$: $N_1(\tau_3) = \emptyset$; $A_1(\tau_3) = \emptyset$; $B_1(\tau_3) = \emptyset$

$m = 2:\ D_{20}(\tau_3) = \mathbf{X}$

$N_2(\tau_3) = \{1, x_1, x_1^2, x_1^3, x_2, x_1x_2\}$; $\delta = 1$:

$A_{21}(\tau_3) = \{(2,4,0),(1,0,2)\}$;

$E_{21}(\tau_3) = \{1, x_1\}$;

$\gamma_{21\tau_3} = x_2 - 4x_1 + 4$

$\xi_{21} = \gamma_{1\tau_3}\gamma_{21\tau_3} = x_2 - 4x_1 + 4;$

$D_{21}(\tau_3) = \{(4,0,0),(2,1,4),(3,0,1),$
$(2,1,3),(1,3,4)\};$

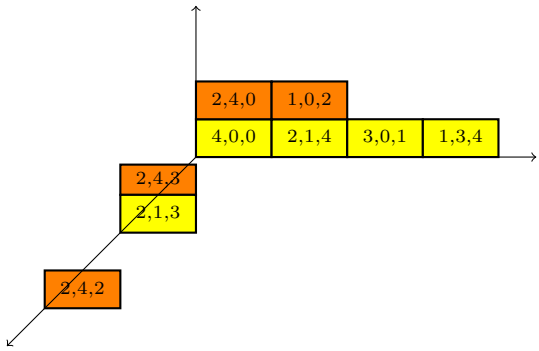
$\delta = 2$:

$A_{22}(\tau_3) = \{(4,0,0),(2,1,4),(3,0,1),$
$(1,3,4)\}$

The terms $vx_m^{d_m-\delta}$ are $1, x_1, x_1^2, x_1^3$ corresponding exactly to P_1, P_2, P_4, P_6.

$E_{22}(\tau_3) = \{1, x_1, x_1^2, x_1^3\};\ \gamma_{22\tau_3} = 2x_2 - x_1^2 + 7x_1 - 12;$

$\xi_{22} = (x_2 - 4x_1 + 4)(2x_2 - x_1^2 + 7x_1 - 12);\ D_{22}(\tau_3) = \emptyset;$



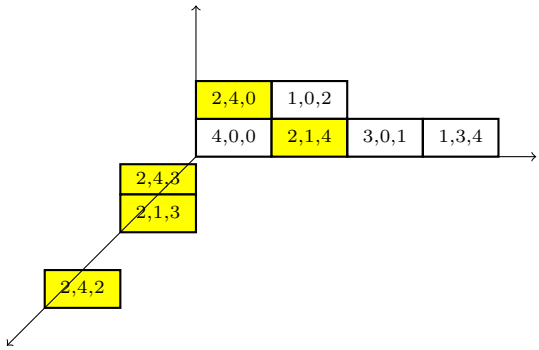
$\tau_4 = x_1x_3$:

$N_1(\tau_4) = \{1\};$

$A_1(\tau_4) = \{(2,1,3)\};$

$B_1(\tau_4) = \{2\}$

$\gamma_{1\tau_4} = (x_1 - 2)$



$m = 2:\ N_2(\tau_4) = \{1\},\ D_{20}(\tau_4) = \{(4,0,0),(3,0,1),(1,3,4),(1,0,2)\}$

$\delta = 1;\ D_{21}(\tau) = D_{20}(\tau);$

$m = 3:\ N_3(\tau_4) = \{1, x_1, x_2, x_1^2, x_3, x_1^3, x_1x_2\};\ \zeta_{m\tau_4} = (x_1 - 2);$

$D_{30}(\tau_4) = \{(4,0,0),(3,0,1),(1,3,4),(1,0,2)\};$

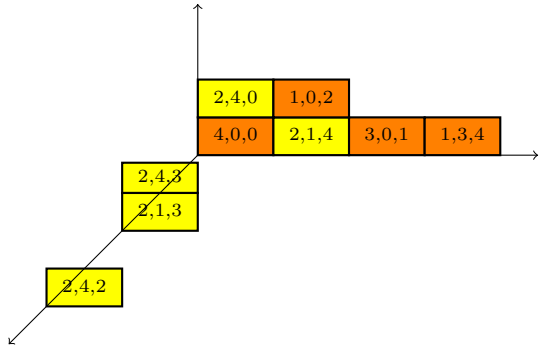
$\delta = 1$:

$A_{31}(\tau_4) = \{(4,0,0),(3,0,1),(1,3,4),$
$(1,0,2)\}$

The terms are $1, x_1, x_1^2, x_1^3, x_2, x_1x_2$, corresponding to $P_1, P_2, P_3, P_4, P_6, P_9$, and P_2, P_3 can be neglected.

$E_{31}(\tau_4) = \{1, x_1, x_1^2, x_2\};\ \gamma_{31}(\tau_4) = 6x_3 - 4x_2 + x_1^2 - x_1 - 12;$

$\xi_{31} = (x_1 - 2)(6x_3 - 4x_2 + x_1^2 - x_1 - 12);\ D_{31}(\tau_4) = \emptyset$ and $\gamma_{3\tau_4} = \gamma_{31}(\tau_4)$.



$\tau_5 = x_2x_3^2:\ N_1(\tau_5) = \emptyset;\ A_1(\tau_5) = \emptyset;\ B_1(\tau_5) = \emptyset$

$m = 2$:

$N_2(\tau_5) = \{1\};$

$D_{20}(\tau_5) = \mathbf{X};$

$\delta = 1$:

$A_{21}(\tau_5) = \{(2,4,2)\};$

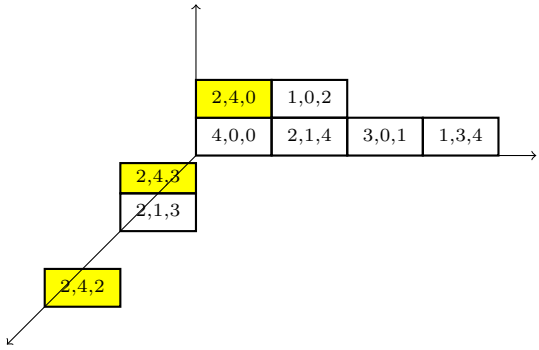
$E_{21}(\tau_5) = \{1\};$

$\gamma_{21\tau_5} = x_2 - 4$

$\xi_{21} = x_2 - 4;$

$D_{21}(\tau_5) = \{(4,0,0),(2,1,4),(3,0,1),$
$(2,1,3),(1,3,4),(1,0,2)\};$

$m = 3:\ \zeta_{3\tau_5} = x_2 - 4;\ D_{30}(\tau_5) = D_{21}(\tau_5);$



$N_3(\tau) = \mathsf{N}(\mathbf{X})$;

$\delta = 1$:

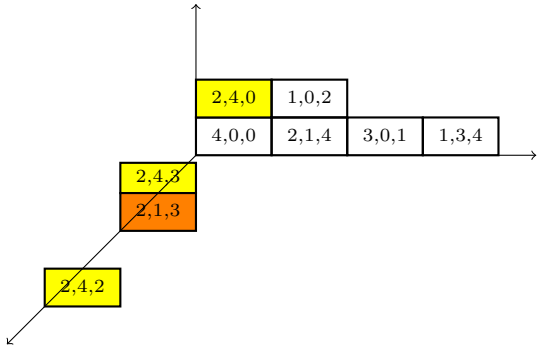
$A_{31}(\tau) = \{(2,1,3)\}$.

$E_{31}(\tau) = \{1\}$;

$\gamma_{21\tau} = x_3 - 3$

$\xi_{31} = (x_2 - 4)(x_3 - 3)$;

$D_{31}(\tau) = \{(4,0,0),(2,1,4),(3,0,1),$
$(1,3,4),(1,0,2)\}$;

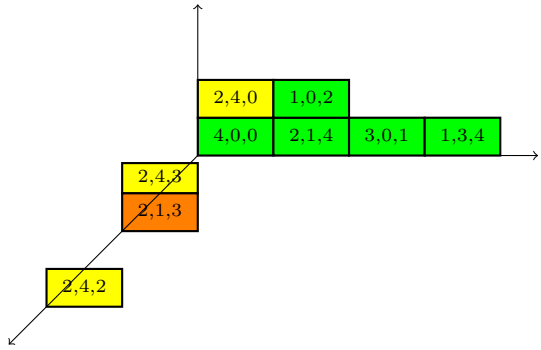


$\delta = 2: A_{32}(\tau) = D_{31}(\tau)$; $E_{32}(\tau) = \{1, x_1, x_1^2, x_1^3, x_2\}$;

$\gamma_{32\tau} = x_3 - 4x_2 - 5x_1^3 + 41x_1^2 - 96x_1 + 48$;

$\xi_{32} = (x_2 - 4)(x_3 - 3)(x_3 - 4x_2 - 5x_1^3 + 41x_1^2 - 96x_1 + 48)$; $D_{32}(\tau) = \emptyset$;

$\gamma_{3\tau} = (x_3 - 3)(x_3 - 4x_2 - 5x_1^3 + 41x_1^2 - 96x_1 + 48)$;



$\tau_6 = x_3^3 :\ N_1(\tau_6) = \emptyset;\ A_1(\tau_6) = \emptyset;\ B_1(\tau_6) = \emptyset$

$m = 2 :\ D_{20}(\tau_6) = \mathbf{X};\ N_2(\tau_6) = \emptyset;$

$\delta = 1 :\ A_{21}(\tau_6) = \emptyset;\ D_{21}(\tau_6) = \mathbf{X};$

$m = 3 :\ D_{30} = \mathbf{X};$

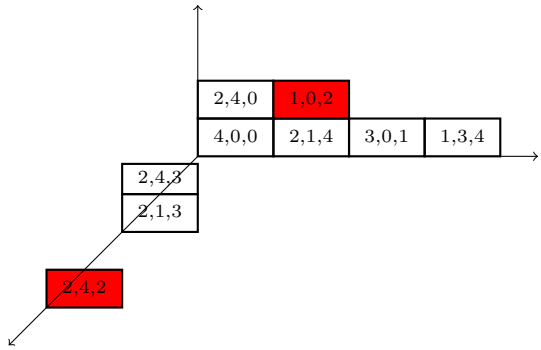
$N_3(\tau_6) = \mathsf{N}(\mathbf{X});$ $\delta = 1$:

$A_{31}(\tau_6) = \{(2, 4, 2)\};$

$E_{31}(\tau_6) = \{1\};$

$\gamma_{31\tau_6} = x_3 - 2;$

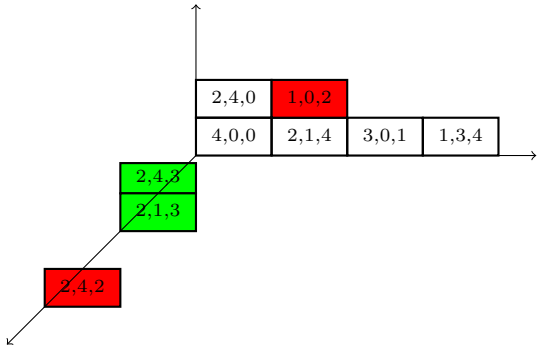
$\xi_{31} = x_3 - 2;$



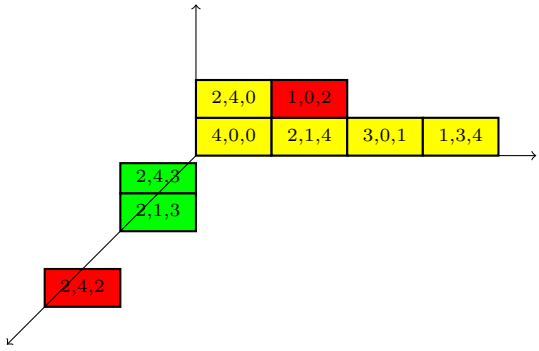
$D_{31}(\tau_6) = \{(4, 0, 0), (2, 1, 4), (2, 4, 0),$
$(3, 0, 1), (2, 1, 3), (1, 3, 4), (2, 4, 3)\};$

$\delta = 2 :\ A_{32}(\tau_6) = \{(2, 1, 3), (2, 4, 3)\};\ E_{32}(\tau_6) = \{1, x_2\};\ \gamma_{32\tau_6} = x_3 - 3;$

$\xi_{32} = (x_3 - 2)(x_3 - 3);\ D_{32} = \{(4, 0, 0), (2, 1, 4), (2, 4, 0), (3, 0, 1), (1, 3, 4)\};$



$\delta = 3:\ A_{33}(\tau_6) = D_{32};\ E_{33}(\tau_6) = \{1, x_1, x_1^2, x_1^3, x_2\};$
$\gamma_{33\tau_6} = 6x_3 + 8x_2 - 5x_1^3 + 35x_1^2 - 54x_1 + 24;$
$\xi_{33} = (x_3 - 2)(x_3 - 3)(6x_3 + 8x_2 - 5x_1^3 + 35x_1^2 - 54x_1 + 24);$
$D_{33}(\tau_6) = \emptyset;$
$\gamma_{3\tau_6} = (x_3 - 2)(x_3 - 3)(6x_3 + 8x_2 - 5x_1^3 + 35x_1^2 - 54x_1 + 24).$



The factorized reduced Groebner basis for $I(\mathbf{X})$ w.r.t. lex is:

$$\mathcal{G}(I(\mathbf{X})) = \Big\{(x_1 - 4)(x_1 - 2)(x_1 - 3)(x_1 - 1), (x_1 - 2)(x_1 - 1)x_2,$$
$$(x_2 - 4x_1 + 4)(2x_2 - x_1^2 + 7x_1 - 12), (x_1 - 2)(6x_3 - 4x_2 + x_1^2 - x_1 - 12),$$
$$(x_2 - 4)(x_3 - 3)(x_3 - 4x_2 - 5x_1^3 + 41x_1^2 - 96x_1 + 48),$$
$$(x_3 - 2)(x_3 - 3)(6x_3 + 8x_2 - 5x_1^3 + 35x_1^2 - 54x_1 + 24)\Big\},$$

while the reduced Groebner basis of $I(\mathbf{X})$ w.r.t. lex is:

$$\begin{gathered}\mathcal{G}'(I(\mathbf{X})) = \Big\{x_1^4 - 10x_1^3 + 35x_1^2 - 50x_1 + 24, x_2x_1^2 - 3x_2x_1 + 2x_2,\\ x_2^2 - 2x_2x_1 - x_2 + 2x_1^3 - 16x_1^2 + 38x_1 - 24, x_3x_1 - 2x_3 - \tfrac{2}{3}x_2x_1 + \tfrac{4}{3}x_2 +\\ + \tfrac{1}{6}x^3 - \tfrac{1}{2}x_1^2 - \tfrac{5}{3}x_1 + 4, x_3^2x_2 - 4x_3^2 - 7x_3x_2 + 28x_3 + \tfrac{8}{3}x_2x_1 +\\ + \tfrac{20}{3}x_2 - \tfrac{16}{3}x^3 + 48x^2 - \tfrac{344}{3}x_1 + 32, x_3^3 - 5x_3^2 + \tfrac{8}{3}x_3x_2 - \tfrac{14}{3}x_3 - \tfrac{16}{9}x_2x_1\\ - \tfrac{40}{9}x_2 + \tfrac{73}{9}x_1^3 - \tfrac{197}{3}x_1^2 + \tfrac{1358}{9}x_1 - 72\Big\},\end{gathered}$$

and it is obtained reducing the polynomials in $\mathcal{G}(I(\mathbf{X}))$, each one w.r.t. the previous ones.

CHAPTER 4

Intermezzo: factorization à la Macaulay.

4.1 Introduction.

As we explained in chapter 3, given a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\}$, the original Axis of Evil algorithm provides a minimal Groebner basis for the zerodimensional radical ideal of these points $I := I(\mathbf{X})$, factorized in a very peculiar way we called "Axis of Evil factorization".
Such a factorization is constructed providing, for each term $\tau \in \mathsf{G}(I)$ a partition $\mathbf{X} = \bigsqcup_{m=1}^{n} \bigsqcup_{\delta=1}^{d_m} \mathsf{S}_{m\delta}(\tau)$ of the points.
As we highlighted in the detailed example of section 3.6, the points are grouped differently at each step: the points in which we have to interpolate the single factors depend on the term $\tau \in \mathsf{G}(I)$ we are considering in the current step of the algorithm.
Moreover, we can notice that in the original Axis of Evil algorithm 5 of chapter 3 some linear factors appearing in the Axis of Evil factorization associated to some terms in $\mathsf{G}(I)$ are

independently computed more than once.
To be more precise, if some factor f appears in the Axis of Evil factorization associated to r terms, then it is computed independently r times.
In this chapter we study the tower structure of points (see 1.4) in order to establish up to what extent it is possible to minimize the number of computed factors.
We will obtain again an Axis of Evil factorization for a minimal Groebner basis of the ideal $I = I(\mathbf{X})$, starting from $\mathbf{X}$ and passing through the computation of both the Groebner escalier $\mathsf{N}(I)$, via some combinatorial algorithm, and of $\mathsf{G}(I)$ from $\mathsf{N}(I)$, via Lazard algorithm 3.2.
We will also show that, in some cases, we can get an Axis of Evil factorization à la Macaulay, in the sense that, if $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{G}(I)$, f_τ is the polynomial in $I(\mathbf{X})$ whose factorization we want to compute and $f_i^{(j)}$ are linear factors with $\mathsf{T}(f_i^{(j)}) = x_i$, then

$$f_\tau = f_1^{(1)} \cdots f_1^{(\alpha_1)} f_2^{(1)} \cdots f_2^{(\alpha_2)} \cdots f_n^{(1)} \cdots f_n^{(\alpha_n)}.$$

Actually, we will show that it is not possible in general.
We will show then that it is possible to construct a similar factorization for some more sets of points, explaining

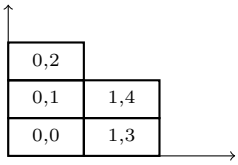
- how to decide whether a set of points admits such a factorization;
- how to get concretely the factorization.

For this factorization, no repeated factors are computed. More precisely, once examined the tower structure associated to $\mathbf{X}$, we exactly know how many factors we need in order to obtain the factorization and which are the corresponding ranges. We only deal with these factors, computing them iteratively on the points.
Anyway, the whole algorithm is not iterative on $\mathbf{X}$, requiring some preprocessing on the points: we need to know all their tower structure before starting the computation.
For this aim, we define another combinatorial method for computing the Groebner escalier $\mathsf{N}(I)$ directly from $\mathbf{X}$, namely the Jumping algorithm, whose aim is to provide a biunivocal correspondence between points and terms in $\mathsf{N}(I)$, taking into account the tower structure.

4.2 First step: back to towers.

In this section, we examine the tower structures of some sets of points in $\mathbf{k}^n$, $n \geq 2$, putting these structure in relation with the Axis of Evil factorization.
Let us start with the case $n = 2$. In two variables, the situation is rather simple.

Indeed, as seen in 2.2.8 *each set of points* $\mathbf{X} \subset \mathbf{k}^2$ *has an unmixed tower structure*. Indeed, it is possible to find out the Groebner escalier by reordering the towers in nonincreasing order by height.
For example, if $\mathbf{X} = \{(0,0),(1,3),(0,1),(0,2),(1,4)\}$ and, as usual, $I = I(\mathbf{X})$, we can get the following unmixed tower structure, associated to $\mathsf{N}(I) = \{1, x_1, x_2, x_1x_2, x_2^2\} \subset \mathbf{k}[x_1, x_2]$



The monomial basis associated to $\mathsf{N}(I)$ is $\mathsf{G}(I) = \{x_1^2, x_1x_2^2, x_2^3\}$, so, if we want to get an Axis of Evil factorization, we have to compute a linear factorization for $f_{x_1^2}$, $f_{x_1x_2^2}$ and $f_{x_2^3}$, such that $\mathsf{T}(f_{x_1^2}) = x_1^2$, $\mathsf{T}(f_{x_1x_2^2}) = x_1x_2^2$ and $\mathsf{T}(f_{x_2^3}) = x_2^3$.
Consider the following lists of polynomials:

$$\Xi_1 = [x_1, x_1 - 1] = [f_1^{(1)}, f_1^{(2)}]$$

$$\Xi_2 = [x_2 - 3x_1, x_2 - 3x_1 - 1, x_2 - 2] = [f_2^{(1)}, f_2^{(2)}, f_3^{(2)}].$$

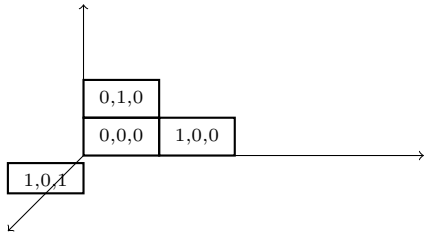
Actually, $f_1^{(1)}, f_1^{(2)}$ come from interpolation on the points corresponding to $1, x_1$, i.e. to the terms of the first x_2-range, whereas $f_2^{(1)}, f_2^{(2)}, f_3^{(2)}$ are interpolated respectively on the points of the first, the second and the third x_2-range.
If we take

$$f_{x_1^2} = x_1(x_1 - 1) = f_1^{(1)} f_1^{(2)}$$

$$f_{x_1x_2^2} = x_1(x_2 - 3x_1)(x_2 - 3x_1 - 1) = f_1^{(1)} f_2^{(1)} f_2^{(2)}$$

$$f_{x_2^3} = (x_2 - 3x_1)(x_2 - 3x_1 - 1)(x_2 - 2) = f_2^{(1)} f_2^{(2)} f_3^{(2)}$$

we obtain an Axis of Evil factorization à la Macaulay for a minimal Groebner basis of $I(\mathbf{X})$.
The case of $n = 3$ is a bit more cumbersome. Indeed for some sets does not exist an unmixed tower structure.
Let us consider a minimal example, i.e. $\mathbf{X} = \{(0,0,0),(1,0,0),(0,1,0),(1,0,1)\}$.
For this, the towers turn out to be mixed regardless the way in which the points are disposed.
For example we can represent the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$ as



The monomial basis associated to $\mathsf{N}(I(\mathbf{X}))$ is $\mathsf{G}(I(\mathbf{X})) = \{x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2\}$.
The fact of having mixed towers actually affects the factorization. Consider the lists of polynomials

$$\Xi_1 = [x_1, x_1 - 1] = [f_1^{(1)}, f_1^{(2)}]$$

$$\Xi_2 = [x_2, x_2 - 1] = [f_2^{(1)}, f_2^{(2)}]$$

$$\Xi_3 = [x_3, x_3 - 1] = [f_3^{(1)}, f_3^{(2)}].$$

We got the polynomials in Ξ_1 interpolating in the points of the first x_2-range. The polynomials in Ξ_2 are obtained by interpolating over the first and the second x_2-range, whereas the ones in Ξ_3 interpolating respectively on the first and the second x_3-range.
We get an Axis of Evil factorization for a minimal Groebner basis of $I(\mathbf{X})$ by

$$f_{x_1^2} = x_1(x_1 - 1) = f_1^{(1)} f_1^{(2)}$$

$$f_{x_1x_2} = x_1x_2 = f_1^{(1)} f_2^{(1)}$$

$$f_{x_2^2} = x_2(x_2 - 1) = f_2^{(1)} f_2^{(2)}$$

$$\mathbf{f_{x_1x_3} = (x_1 - 1)x_3 = f_1^{(2)} f_3^{(1)}}$$

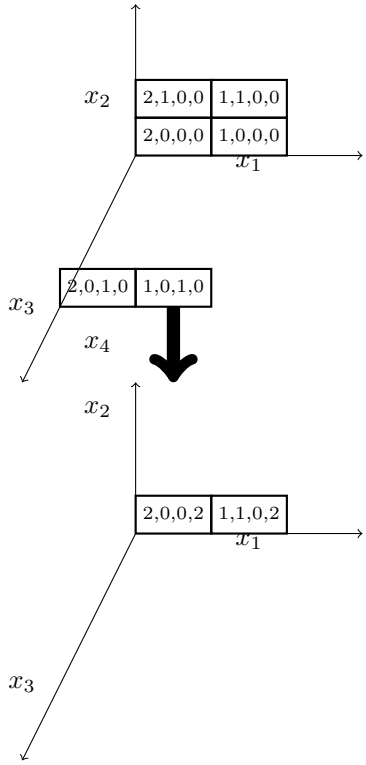
$$f_{x_2x_3} = x_2x_3 = f_2^{(1)} f_3^{(1)}$$

$$f_{x_3^2} = x_3(x_3 - 1) = f_3^{(1)} f_3^{(2)}.$$

Notice that $f_{x_1x_3} = (x_1 - 1)x_3 = f_1^{(2)} f_3^{(1)}$ that we have highlighted on purpose, is not really Macaulay-like, since we do not take $f_1^{(1)}$. Anyway, it is not so different and, mainly, the factors in Ξ_1, Ξ_2, Ξ_3 are enough to get the whole factorization.
Let us consider the case $n = 4$. Here, the situation can be even more complicated. Clearly there can be sets which cannot have unmixed towers. Look at the following simple set:

$$\mathbf{X} = \{(2,0,0,0), (1,0,0,0), (2,1,0,0), (1,1,0,0), (2,0,1,0), (1,0,1,0), (2,0,0,2), (1,1,0,2)\}.$$

We can represent its tower structure as:



and the Groebner escalier is $\mathsf{N}(I(\mathbf{X})) = \{1, x_1, x_2, x_1x_2, x_3, x_1x_3, x_4, x_1x_4\}$. Consider now the sets

$$\Xi_1 = [x_1 - 2, x_1 - 1] = [f_1^{(1)}, f_1^{(2)}]$$
$$\Xi_2 = [x_2, x_2 - 1] = [f_2^{(1)}, f_2^{(2)}]$$
$$\Xi_3 = [x_3, x_3 - 1] = [f_3^{(1)}, f_3^{(2)}]$$
$$\Xi_4 = [x_4, x_4 - 2] = [f_4^{(1)}, f_4^{(2)}],$$

obtained as in the examples above. The term x_2x_4 belongs to the monomial basis associated to $\mathsf{N}(I(\mathbf{X}))$. If we want to find a factorization for $f_{x_2x_4}$ we first take $f_4^{(1)}$, vanishing at all the points of the first x_4-range, but none of the linear factors in Ξ_2 vanishes at both $(2, 0, 0, 2)$ and $(1, 1, 0, 2)$: the factors in Ξ_2 are not enough to provide the whole Axis of Evil factorization.

4.3 Second step: the Jumping algorithm.

The *Jumping algorithm* places itself in the context introduced in chapter 2, where combinatorial methods to compute the (finite) Groebner escalier of a zerodimensional radical ideal

are defined.
This algorithm configures itself as an alternative to the methods already proposed and, in particular, it shows a strong relationship with Felszeghy-B. Ráth-Rónyai Lex Game of which it is, to all intents and purposes, a variation.
Indeed, it exploits again the idea of ordering the points of a given finite set in a trie structure, but it proceeds differently in its concrete construction.
The result of this new construction is again a one to one correspondence between the points in the set and the terms constituting the Groebner escalier, but the tower reordering is taken into account while constructing the Groebner escalier.
We can interpret the Jumping algorithm as an *interpolation oriented Lex Game*, since in some case it can help to produce an Axis of Evil factorization à la Macaulay.
We explain now the algorithm in detail.
Consider a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\} \subseteq \mathbf{k}^n$.
As usual, we denote by $I = I(\mathbf{X})$ the (zerodimensional radical) ideal associated to $\mathbf{X}$ and $\mathsf{N}(I) = \mathsf{N}(I(\mathbf{X}))$ its Groebner escalier. In order to construct $\mathsf{N}(I)$, the algorithm

a) constructs a trie $\mathfrak{T}(\mathbf{X})$ associated to $\mathbf{X}$, we name *children trie*, a variation of the point trie by Felszeghy-B. Ráth-Rónyai ;

b) constructs the lex trie as in the Lex Game.

As we had already studied step $b)$ in chapter 2, we only deal with step $a)$.
Therefore, we equip again $\mathbf{k}^n$ with the equivalence relation we denoted by $=$
$a = (a_1, ..., a_n) = (b_1, ..., b_n) = b$ if $a_i = b_i$, $\forall i \in \{1, ..., n\}$ (see 2.5).
Taken then our points $P_1, ...P_S \in \mathbf{X} \subseteq \mathbf{k}^n$, we define the equivalence classes of $\pi_i(P_j), i = 1, ..., n,\ j = 1, ..., S$, calling them Σ_i and representing them as sets containing the indices of the points in the class, instead of taking trace of the points.
Clearly $\Sigma_0 = \{\{1, ..., S\}\}$, $\Sigma_n = \{\{1\}, ..., \{S\}\}$, $|\Sigma_n| = S$.
Then, we construct a trie whose vertices are labeled with the elements $\Sigma_{i,k} \in \Sigma_i$, for $i = 1, ..., n$, $k = 1, ..., |\Sigma_i|$. We set an edge from $\Sigma_{i,k} \in \Sigma_i$ to $\Sigma_{i+1,h}, \in \Sigma_{i+1}$ when $\Sigma_{i+1,h} \subseteq \Sigma_{i,k}$ and we label it with the $(i+1)$-th coordinate of the points in Σ_{i+1}.
As a second step, we have to put an ordering on the classes. More precisely, we examine the levels from $n-2$ to 0 and we order the children of each node in the level under consideration. We perform the steps described below.

a) For each node a at level $n-2$, we order its children $b_1, ..., b_{h_1}$, according to the number of leaves depending on them. If c_i nodes depend on b_i and and c_j nodes depend on b_j and $c_i > c_j$ for $i, j \in \{1, ..., h_1\}$, then we pose b_i on the left of b_j. Possibly, there can

be nodes from which depend the same number of leaves. In this case, their mutual position is indifferent.
While making this ordering, we keep track in a list L of the number of leaves depending on each child. We do not allow repetitions in L, so if some number occurs more than once, we keep track of it together with the number of time it occurs.

b) For each node a at level $n-3$ we order its children $b_1, ..., b_{h_2}$, associating to each of them a list containing the number of children and the list obtained in the previous step, separating with a marker the two objects. Then we compare the lists. We put on the left a node if in the corresponding list we find a bigger number or the same number occurring more times. If two lists are equal, the mutual position of the associated children is indifferent.
While ordering the nodes, we prepare a new list, analogous to the list L of a) in which, for each block identified by the markers, we write down the numbers we examine, again equipped with the number of times they occur.

c) For each node a of level i, we order its children. Each of them is equipped as before with a list, containing the number of nodes depending on it and the list obtained in the previous step (always equipped with markers). Then we compare the lists as before, keeping track again of the data in order to use them in the next step.

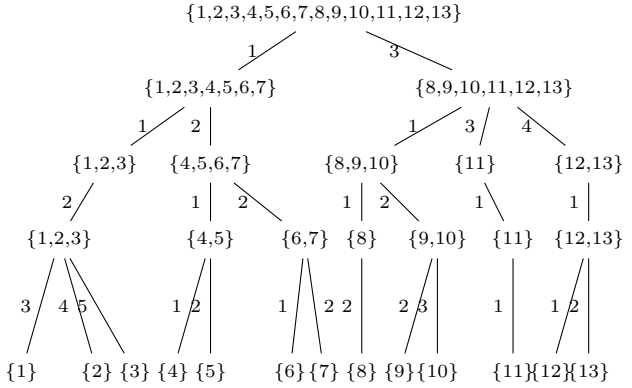
At the end we obtain an ordering on the classes in the trie.

Example 4.3.1. Consider the set

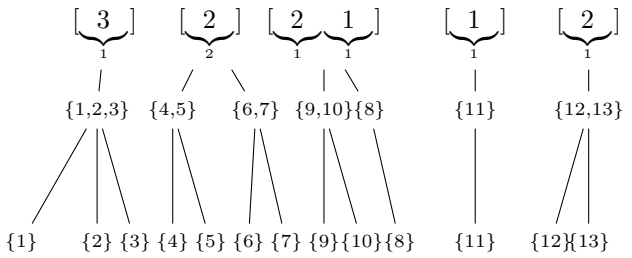
$$\mathbf{X_1} = \{(1,1,2,3),(1,1,2,4),(1,1,2,5),(1,2,1,1),(1,2,1,2),(1,2,2,1),(1,2,2,2),$$
$$(3,1,1,2),(3,1,2,2),(3,1,2,3),(3,3,1,1),(3,4,1,1),(3,4,1,2)\}.$$

For this set, we compute the equivalence classes Σ_i , $i = 0, ..., 4$.
The first class, Σ_0, is trivial: $\Sigma_0 = \{\{1,2,3,...,13\}\}$
For Σ_1, we observe that $\pi_1(\mathbf{X_1}) = \{1,3\}$, so $|\Sigma_1| = 2$. Its two elements are the set $\Sigma_{1,1} = \{1,2,3,4,5,6,7\}$ and the set $\Sigma_{1,2} = \{8,9,10,11,12,13\}$.
We have $\Sigma_1 = \{\{1,2,3,4,5,6,7\},\{8,9,10,11,12,13\}\}$.
For Σ_2 we proceed in the same way and we put the points starting with the couples $(1,2)$, $(1,1)$, $(3,1)$, $(3,4)$, $(3,3)$:
$\Sigma_2 = \{\{1,2,3\},\{4,5,6,7\}\{8,9,10\},\{11\},\{12,13\}\}$.
Constructing Σ_3, we are setting the points starting with these 3-tuples: $(3,1,2)$, $(3,1,1)$, $(3,4,1)$, $(3,3,1)$, $(1,2,1)$, $(1,2,2)$, $(1,1,2)$.

Consequently, we get $\Sigma_3 = \{\{1,2,3\},\{4,5\},\{6,7\},\{8\},\{9,10\},\{11\},\{12,13\}\}$.
Finally, we write down the single points:
$\Sigma_4 = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\},\{12\},\{13\}\}$. Up to now, we did not yet order the classes, so we get:

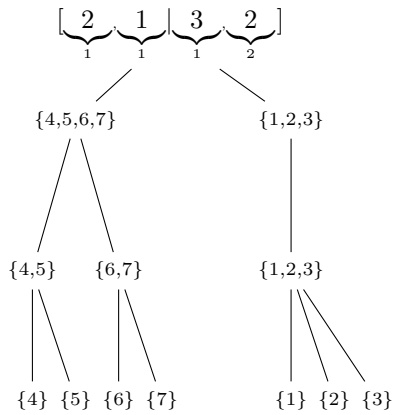


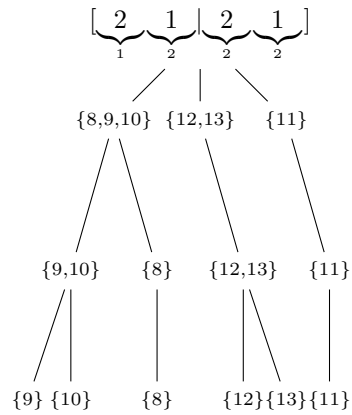
Now we order the classes.
Since the node $\{1,2,3\}$ at level 3 is the only one depending on the node $\{1,2,3\}$ at level 2 we do not have to order it. We only keep track that 3 leaves depend on it.
We have to order the nodes $\{4,5\},\{6,7\}$ at level 3 (depending on the node $\{4,5,6,7\}$ at level 2). Since two leaves depend on each of them, their mutual position is indifferent. We keep track of the fact that 2 leaves occur twice.
Consider then $\{8\},\{9,10\}$ (depending on the level 2 node $\{8,9,10\}$). The set $\{9,10\}$ goes on the left of $\{8\}$. Indeed $\{9,10\}$ has 2 leaves, whereas $\{8\}$ has only one. We keep track of the numbers of leaves, which are $2,1$, each one appearing once. The sets $\{11\}$ and $\{12,13\}$ have not to be ordered (as it was for the first set). We only keep track of the leaves. So we have:



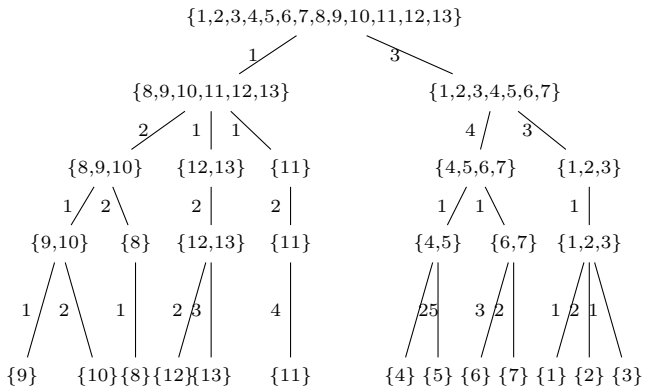
Then, for each node at level 2, we attach to the corresponding lists obtained before and displayed in the first row of the above picture also the number of children, obtaining:

$[1|\underbrace{3}_{1}]$
$[2|\underbrace{2}_{2}]$,
referring to the children of $\{1,2,3,4,5,6\}$ and
$[2|\underbrace{2}_{1},\underbrace{1}_{1}]$
$[1|\underbrace{1}_{1}]$
$[1|\underbrace{2}_{1}]$,
referring to the children of $\{8,9,10,11,12,13\}$.
Then we compare the lists and we get





The lists we have to compare to order $\{1,2,3,4,5,6,7\}$ and $\{8,9,10,11,12,13\}$ are
$[2|\underbrace{2}_{1},\underbrace{1}_{1}|\underbrace{3}_{1},\underbrace{2}_{2}]$
and
$[3|\underbrace{2}_{1}\underbrace{1}_{2}|\underbrace{2}_{2}\underbrace{1}_{2}]$.
Since $3 > 2$ we get



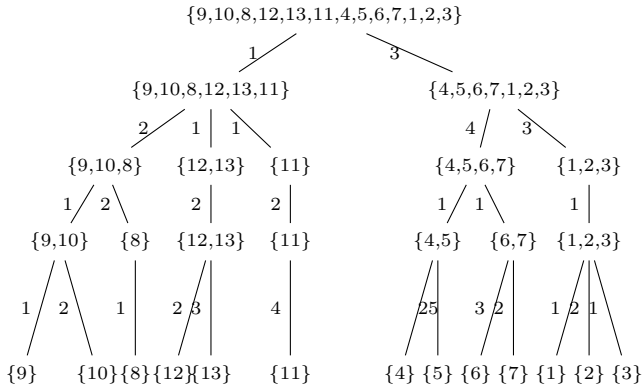
After computing and ordering $\Sigma_1, ..., \Sigma_n$ as explained above, we consider Σ_n. This class is composed of singletons, since the elements of $\mathbf{X}$ are all distinct by hypothesis.
We report on $\Sigma_{n-1}, ..., \Sigma_0$ the ordering of the points induced by the order of the singletons in Σ_n. This means that we *reorder the points in the sets composing the classes*.

Example 4.3.2. Referring to example 4.3.1, we have $n = 4$ and we reorder the points according to
$\Sigma_4 = \{\{9\}, \{10\}, \{8\}, \{12\}, \{13\}, \{11\}, \{4\}, \{5\}, \{6\}, \{7\}, \{1\}, \{2\}, \{3\}\}$, obtaining
$\Sigma_0 = \{\{9, 10, 8, 12, 13, 11, 4, 5, 6, 7, 1, 2, 3\}\}$
$\Sigma_1 = \{\{9, 10, 8, 12, 13, 11\}, \{4, 5, 6, 7, 1, 2, 3\}\}$
$\Sigma_2 = \{\{9, 10, 8\}, \{12, 13\}, \{11\}, \{4, 5, 6, 7\}, \{1, 2, 3\}\}$
$\Sigma_3 = \{\{9, 10\}, \{8\}, \{12, 13\}, \{11\}, \{4, 5\}, \{6, 7\}, \{1, 2, 3\}\}$
$\Sigma_4 = \{\{9\}, \{10\}, \{8\}, \{12\}, \{13\}, \{11\}, \{4\}, \{5\}, \{6\}, \{7\}, \{1\}, \{2\}, \{3\}\}$.

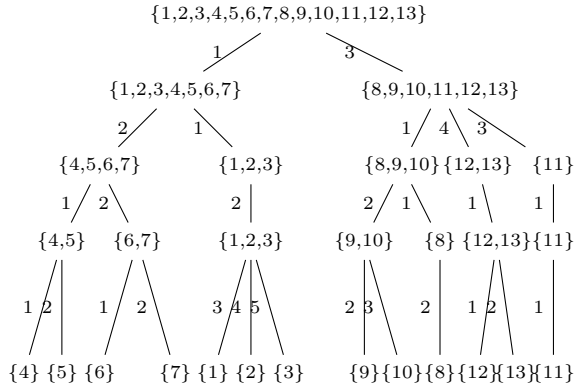
Definition 4.3.3. The *children trie* $\mathfrak{T}(\mathbf{X})$ of a finite set of distinct points $\mathbf{X}$ is the point trie associated to classes ordered w.r.t. the rules explained above.

The children trie $\mathfrak{T}(\mathbf{X})$ is such that $ht(\mathfrak{T}(\mathbf{X})) = n$. Clearly, we have defined a biunivocal correspondence between the points in the given set $\mathbf{X}$ and the paths from the root to the leaves in the tree.

Example 4.3.4. Consider again the set $\mathbf{X_1}$ of examples 4.3.1, 4.3.2. We draw again the associated children trie:



which has a strong link with the Lex Game point trie. Anyway, both the ordering of the classes and the internal order of elements in the classes are different.
Namely, the Lex Game point trie is:

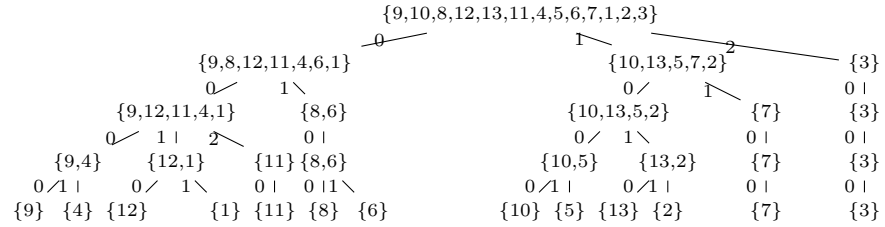


which turns out to be different from the children trie displayed above.

Once we have computed the children trie $\mathfrak{T}(\mathbf{X})$, we only have to perform the lex trie algorithm on $\mathfrak{T}(\mathbf{X})$ in order to determine the Groebner escalier $\mathsf{N}(I)$.

Example 4.3.5. Referring to the set $\mathbf{X_1}$ of example 4.3.4, we perform the lex trie construction. The first set is:
$v_0 = \{9, 10, 8, 12, 13, 11, 4, 5, 6, 7, 1, 2, 3\}$.
We set $h = 1$, so we iterate on Σ_3, getting
$v_{0,0} = \{9, 8, 12, 11, 4, 6, 1\} =: v_0$
$v_{0,1} = \{10, 13, 5, 7, 2\} =: v_1$
$v_{0,2} = \{3\} =: v_2$.

For $h = 2$, we perform the iteration on Σ_2:
$v_{0,0} = \{9, 12, 11, 4, 1\} =: v_0$
$v_{0,1} = \{8, 6\} =: v_1$
$v_{1,0} = \{10, 13, 5, 2\} =: v_2$
$v_{1,1} = \{7\} =: v_3$
$v_{2,0} = \{3\} =: v_4$.
For $h = 3$, the iteration on Σ_1 produces
$v_{0,0} = \{9, 4\} =: v_0$
$v_{0,1} = \{12, 1\} =: v_1$
$v_{0,2} = \{11\} =: v_2$
$v_{1,0} = \{8, 6\} =: v_3$
$v_{2,0} = \{10, 5\} =: v_4$
$v_{2,1} = \{13, 2\} =: v_5$
$v_{3,0} = \{7\} =: v_6$
$v_{4,0} = \{3\} =: v_7$
Finally, for $h = 3$, the iteration on Σ_0 gives
$v_{0,0} = \{9\} =: v_0$
$v_{0,1} = \{4\} =: v_1$
$v_{1,0} = \{12\} =: v_2$
$v_{1,1} = \{1\} =: v_3$
$v_{2,0} = \{11\} =: v_4$
$v_{3,0} = \{8\} =: v_5$
$v_{3,1} = \{6\} =: v_6$
$v_{4,0} = \{10\} =: v_7$
$v_{4,1} = \{5\} =: v_8$
$v_{5,0} = \{13\} =: v_9$
$v_{5,1} = \{2\} =: v_{10}$
$v_{6,0} = \{7\} =: v_{11}$
$v_{7,0} = \{3\} =: v_{12}$
So the output trie is



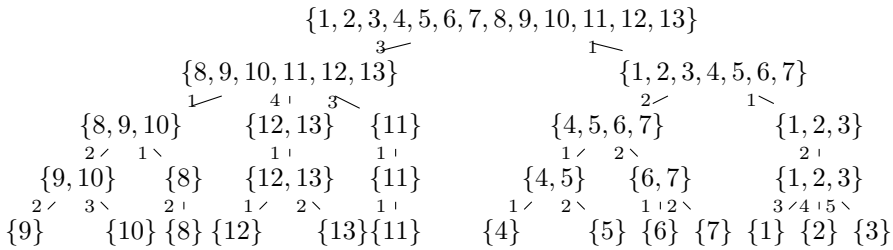
The correctness of the algorithm follows from the one of the Lex Game (see chapter 2). For each $h = 1, ..., n$, at level $n - h$, the points in the same class have at least the first $n - h$ coordinates in common, but we have already settled the corresponding powers of $x_{n-h+2}, ..., x_n$. When we examine h and $n - h$ in the lex trie construction, we settle the powers of x_{n-h+1}, looking at the number of points with the same first $n - h$ coordinates and whose corresponding terms have the same powers of $x_{n-h+2}, ..., x_n$.
Thanks to the children trie construction, while browsing the points, the first points we take into account are those corresponding to higher towers in the subsequent variable (and in case of equality, to the bigger number of high towers in such a variable). If we get again an equality, the comparison passes to the next variable.
This way, we are taking into account the tower reordering, trying to avoid shifts, when it is possible.
Notice that the reordering of the points in the single classes is crucial, as it is shown in the following example.

Example 4.3.6. Take, as usual, the set
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$
$(3,1,1,2), (3,1,2,2), (3,1,2,3), (3,3,1,1), (3,4,1,1), (3,4,1,2)\}$,
and compute the sets Σ_i, taking care to order the equivalence classes as in the Jumping algorithm, but *not reordering the points in the classes*.

$\Sigma_0 = \{\{1, 2, 3, ..., 13\}\}$
$\Sigma_1 = \{\{8, 9, 10, 11, 12, 13\}, \{1, 2, 3, 4, 5, 6, 7\}\}$.
The class starting with 3 is put before the one starting with 1.
$\Sigma_2 = \{\{8, 9, 10\}, \{12, 13\}, \{11\}, \{4, 5, 6, 7\}, \{1, 2, 3\}\}$.

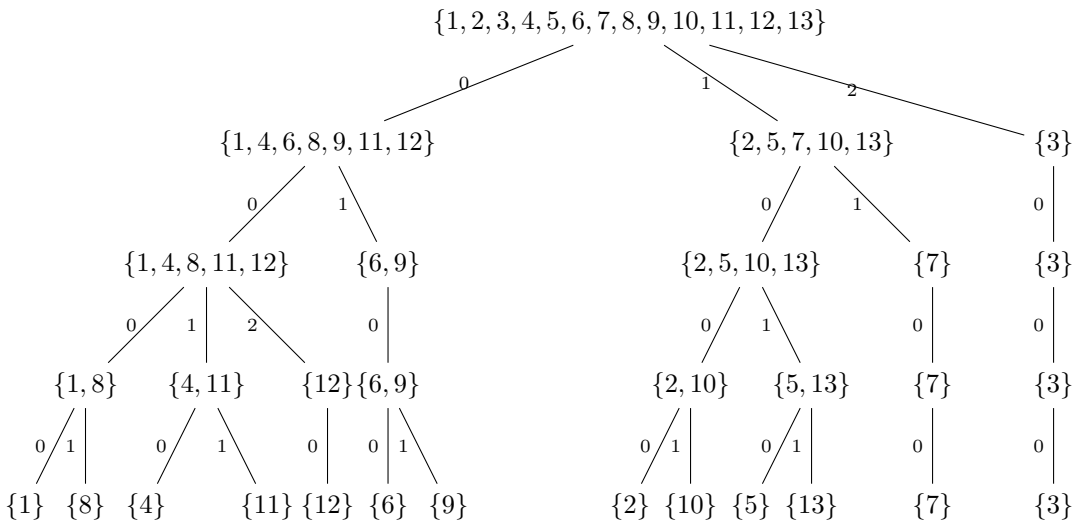
We put now in order the following couples:
$(3,1), (3,4), (3,3), (1,2), (1,1)$.
$\Sigma_3 = \{\{9, 10\}, \{8\}, \{12, 13\}, \{11\}, \{4, 5\}, \{6, 7\}, \{1, 2, 3\}\}$. The 3-tuples are:
$(3,1,2), (3,1,1), (3,4,1), (3,3,1), (1,2,1), (1,2,2), (1,1,2)$.
$\Sigma_4 = \{\{9\}, \{10\}, \{8\}, \{12\}, \{13\}, \{11\}, \{4\}, \{5\}, \{6\}, \{7\}, \{1\}, \{2\}, \{3\}\}$

The classes are now in the order set by the jumping algorithm and the tree has exactly the same shape, even if it contains the points in a different order (see example 4.3.4).

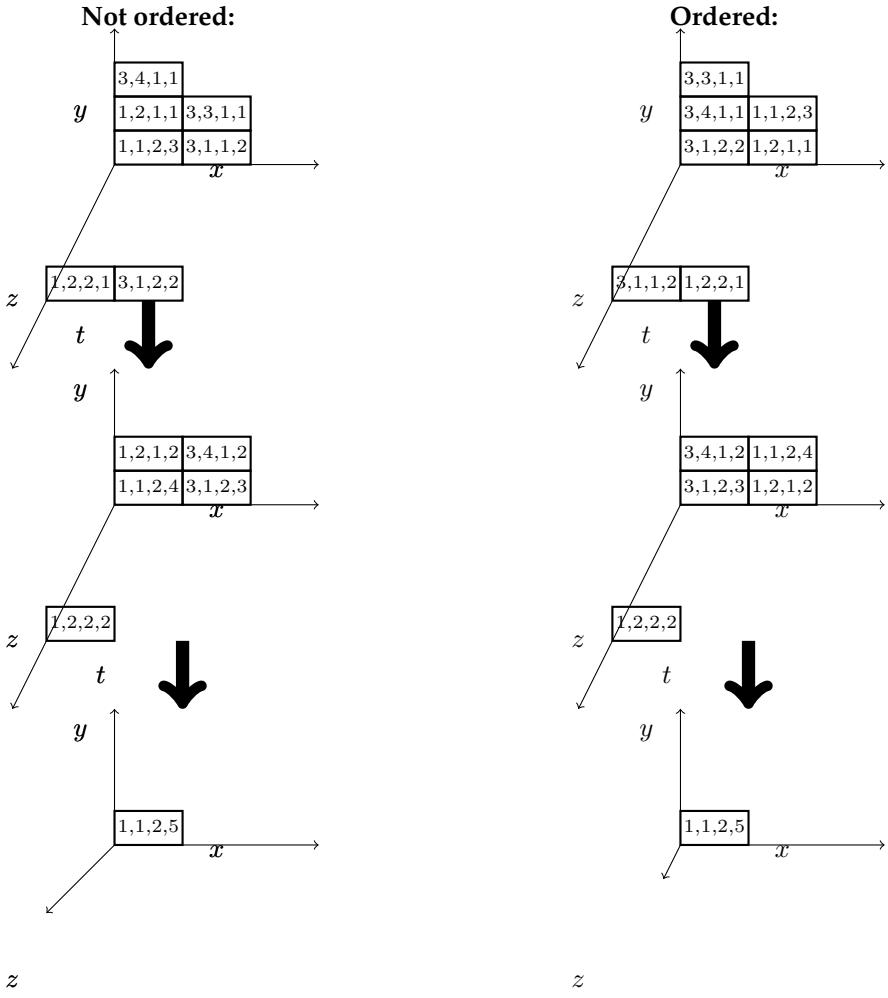


The root of the lex trie is $v_0 = \{1, 2, 3, 4, 5, 6, 7, , 8, 9, 10, 11, 12, 13\}$.
For $h = 1$, we iterate on Σ_3, so
$v_{0,0} = \{9, 8, 12, 11, 4, 6, 1\} =: v_0$
$v_{0,1} = \{10, 13, 5, 7, 2\} =: v_1$
$v_{0,2} = \{3\} =: v_2$.
For $h = 2$, performing an iteration on Σ_2 we get
$v_{0,0} = \{8, 12, 11, 4, 1\} =: v_0$
$v_{0,1} = \{9, 6\} =: v_1$
$v_{1,0} = \{10, 13, 5, 2\} =: v_2$
$v_{1,1} = \{7\} =: v_3$
$v_{2,0} = \{3\} =: v_4$.
For $h = 3$, iterating on Σ_1 we get
$v_{0,0} = \{8, 1\} =: v_0$
$v_{0,1} = \{11, 4\} =: v_1$
$v_{0,2} = \{12\} =: v_2$
$v_{1,0} = \{9, 6\} =: v_3$
$v_{2,0} = \{10, 2\} =: v_4$
$v_{2,1} = \{13, 5\} =: v_5$
$v_{3,0} = \{7\} =: v_6$
$v_{4,0} = \{3\} =: v_7$.
Finally for $h = 4$, we have an iteration on Σ_0, so
$v_{0,0} = \{1\} =: v_0$
$v_{0,1} = \{8\} =: v_1$
$v_{1,0} = \{4\} =: v_2$
$v_{1,1} = \{11\} =: v_3$
$v_{2,0} = \{12\} =: v_4$
$v_{3,0} = \{6\} =: v_5$

$v_{3,1} = \{9\} =: v_6$
$v_{4,0} = \{2\} =: v_7$
$v_{4,1} = \{10\} =: v_8$
$v_{5,0} = \{5\} =: v_9$
$v_{5,1} = \{13\} =: v_{10}$
$v_{6,0} = \{7\} =: v_{11}$
$v_{7,0} = \{3\} =: v_{12}$.
The final trie is

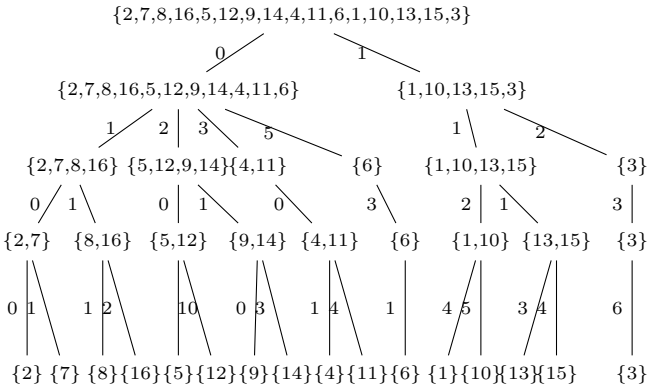


In conclusion, if we do not order the points we obtain another biunivocal correspondence. We represent it below, by displaying the associated tower structure:

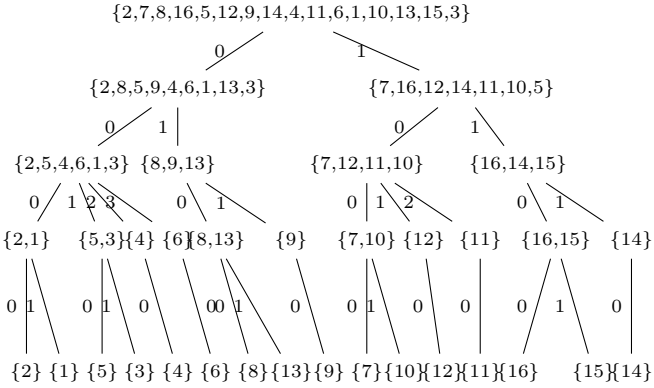


Let us see an example of unmixed towers.

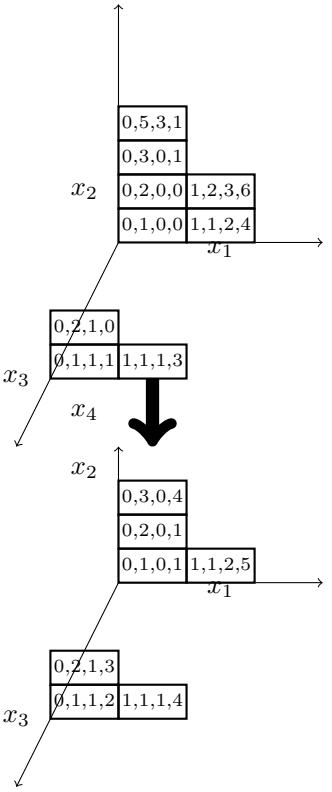
Example 4.3.7. Consider the set $\mathbf{X} = \{(1,1,2,4),(0,1,0,0),(1,2,3,6),(0,3,0,1),(0,2,0,0),(0,5,3,1),(0,1,0,1),(0,1,1,1),(0,2,1,0),(1,1,2,5),(0,3,0,4),(0,2,0,1),(1,1,1,3),(0,2,1,3),(1,1,1,4),(0,1,1,2)\} = \{P_1,...,P_{16}\}$.
The associated children trie is



Performing as usual, we get the lex trie below



and the following tower structure, that is unmixed.



4.4 Third step: Axis of Evil Macaulay factorization.

In this section, denoting as usual by $\mathbf{X}$ a finite set of distinct points and by $I = I(\mathbf{X})$ the zerodimensional radical ideal of points, we try to compute an Axis of Evil factorization of a lexicographical minimal Groebner basis of I, so that it is as similar as possible to the factorization à la Macaulay examined in 4.2 and minimizing the number of factors to compute.
For the original Axis of Evil algorithm, we pointed out that the method used in order to construct the Groebner escalier does not affect the correctness of the algorithm (3.4.20).
In this case, we suppose to employ always the Jumping algorithm in order to properly pass from the points in $\mathbf{X}$ to the terms in the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$.
Take the set $\mathbf{X}$ and apply the Jumping algorithm, obtaining the Groebner escalier $\mathsf{N}(I(\mathbf{X}))$ ordered in the lex trie, but also keeping stored in memory the children trie.
We denote by

$$\Phi_{Jumping} : \mathbf{X} \to \mathsf{N}(I(\mathbf{X}))$$

$$P_i \mapsto \tau_i,$$

the biunivocal correspondence provided by the Jumping algorithm.
Notice that we have a biunivocal correspondence between points in $\mathbf{X}$ and terms in the Groebner escalier, we consider $\mathbf{X}$ reordered by the algorithm above in such a way that $P_i \leftrightarrow \tau_i$ and that the τ_i are in increasing order w.r.t. lex.
For brevity's sake, we employ the notation $\mathbf{X_j} = \{P_1, ..., P_j\}$ and $\mathsf{N}_j = \mathsf{N}(I(\mathbf{X_j}))$.
In order to compute the required factorization, we proceed as follows.

1. The first term is $\tau_1 = 1$ and it corresponds to a point $P_1 = (a_{1,1}, ..., a_{1,n})$[1].
2. Construct n lists Ξ_i, $i = 1, ..., n$ [2], each one containing the factor $x_i - a_{1,i}$, $i = 1, ..., n$.
3. Construct 2 list L_1, H_1 containing n entries equal to 1.
4. Set $\mathsf{G}_1 = \{x_1, ..., x_n\}$
5. For $\tau_j = x_1^{j_1} \cdots x_n^{j_n} \in \mathsf{N}(I(\mathbf{X}))$, $j = 2, ..., n$ repeat steps 6-9
6. Construct lists $L_j = L_{j-1}$, $H_j = [\underbrace{j, ..., j}_{n \text{ times}}]$.
7. Compute the minimal monomial basis G_j associated to N_j. The idea is to perform for each point one step of Lazard algorithm (c.f. section 3.2). Referring to the explanation given of Lazard's algorithm it essentially means removing τ_j from the current basis and inserting $x_1\tau_j, ..., x_n\tau_j$, possibly incrementing the number associated to them if they already appeared in the basis: the elements of G_j are the ones appearing as many times as the number of variables dividing them.
8. If $j_n = 0$ compute the triangular polynomial associated to the corresponding P_j w.r.t $\mathbf{X_{j-1}}$, exactly as performed in Moeller algorithm (see section 1.6).
   More precisely:
   - associate to τ_j the corresponding linear factors, via the *Association procedure* described in (2) below, and multiply them, obtaining a polynomial f_j, such that $T(f_j) = \tau_j$;
   - the triangular polynomial is $q_j = \frac{1}{f_j(P_j)} f_j$.

   Otherwise, if $j_n \neq 0$ go directly to the next step.

[1] This is the base case for our algorithm, since $I(\{P_1\}) = (x_1 - a_{1,1}, ..., x_n - a_{1,n})$, which is "naturally factorized".

[2] One list for each variable: they will contain all the necessary linear factors in order to find the required minimal Groebner basis.

9. Let $x_h = \max(\tau_j)$. Perform a sort of BFS[3] on the children trie starting from the root Σ_0, from level 1 to n, namely

   - consider the first point P_l of $\Sigma_{1,k}$ $\forall \Sigma_{1,k} \in \Sigma_1$[4];
   - if $P_l = P_j$, set $L_j[1] = k$ and $\Sigma_0 = \Sigma_{1,k}$ as new root and repeat the horizontal reading on the subtree whose root is $\Sigma_{1,k}$, using the projection π_2 to compare the points;
   - if $P_l \neq P_j$ and $\pi_1(P_l) = \pi_1(P_j)$ set $L_j[1] = L_l[1]$, $H_j[1] = H_l[1]$ and $\Sigma_0 = \Sigma_{1,k}$ as new root and repeat the horizontal reading on the subtree whose root is $\Sigma_{1,k}$, using the projection π_2 to compare the points;
   - if $P_l \neq P_j$ and $\pi_1(P_l) \neq \pi_1(P_j)$ continue the horizontal reading with $\Sigma_{1,k+1}$.

10. For each x_s, $s < h$, take the x_s-range of P_j and of the point P_l found for level l in the BFS. For each point P_m in the x_s-range of P_j check whether there is a point in the x_s-range of P_l, sharing the first s coordinates with P_m. If it is not possible to find it stop the execution. The factorized polynomials f_σ, $\sigma \in \mathsf{G}(I)$ of the minimal Groebner basis we are looking for are computable via the association procedure described below, for each $\sigma \in \mathsf{G}(I)$, $\sigma < \tau_j$. For the other ones, we have to switch to the first Axis of Evil algorithm 5[5].

11. For each x_l, $l \geq h$, we update the factors in the variable x_l as follows.

    - If $\tau_j = x_h^m$, i.e. it is a pure power, then add $x_h - a_{j,h}$ to the linear factors whose leading term is x_h. Then interpolate the factors in $x_{h+1}, ..., x_n$ associated to the ranges containing P_j, using the *interpolation algorithm* (1) described below. Set $L_j[h] = |\Xi_h|$.
    - If τ_j, is not a pure power, use the interpolation algorithm (1) on the last factors in x_i, $i = h..., n$ and set $L_j[h] = |\Xi_h|$.

12. Associate to each term in $\mathsf{G}_m = \mathsf{G}(I(\mathbf{X}))$ a factorized polynomial via the Association procedure (2) and return the result.

Let us now examine the subroutines needed to perform the algorithm.
(1) *Interpolation algorithm*
for the point P_j, $j = 2, ..., m$ and a generic factor p, letting $\Phi_{Jumping}(P_j) = \tau_j$ and $\max(\tau_j) =$

[3] In the sense that the nodes of the trie are examined horizontally, see [61].
[4] We are reading the first point for each set labeling the nodes in the first level of the children trie.
[5] This is not a problem since the factorization of algorithm 5 is computed independently for each term in the monomial basis (see 3.4.10).

x_h. Denote by P_s the point found for level $h-1$ in the BFS and by q_s its associated triangular polynomial.

i) Compute $v = ev_{P_j}(p)$;

ii) If $T(p) \leq \tau_j$ assign to p the value $p - vq_s$.

iii) If $T(p) > \tau_j$ assign to p the value $p - vq_j$;

(2) *Association procedure*
for a term $\sigma = x_1^{j_1} \cdots x_h^{j_h}$.

1) store the first j_h factors in Ξ_j;

2) set $\sigma' = x_h^{j_h}$

3) consider $P_l = \Phi^{-1}_{Jumping}(\sigma')$;

4) compute $x_i = \max(\frac{\sigma}{\sigma'})$;

5) store $\Xi_i[L_l[i]]$;

6) set $\sigma' = \sigma' x_i$;

7) set $P_l = \Phi^{-1}_{Jumping}(\sigma')$;

8) repeat steps from $4)$ to $7)$ until $\sigma' = \sigma$.

Remark 4.4.1. If $\tau_j \in \mathsf{N}(I)$, $\max(\tau_j) = x_n$, then we can omit the computation of the corresponding triangular polynomial q_j. Indeed, $\mathsf{T}(q_j) = \tau_j \geq x_n$, so this q_j can never be used to interpolate the linear factors, since if so, q_j can modify the leading term.

Remark 4.4.2. Notice that, unlike the originary Axis of Evil algorithm, this version performs a loop on $\mathbf{X}$.
Anyway, it cannot be really iterative on the points since we need to have performed the Jumping algorithm as a preprocessing.

Remark 4.4.3. Given $\mathbf{X} = \{P_1, ..., P_S\}$ (ordered via the jumping algorithm), consider subsets $\mathbf{Y} = \{P_1, ..., P_t\} \subseteq \mathbf{X}$, $t \leq S$.
We point out that our algorithm, being iterative on the points, can also produce the linear factorization for a minimal Groebner basis of the vanishing ideal of every such $\mathbf{Y}$. If such factorization is needed we can show it, computing $\mathsf{G}(I(\mathbf{Y}))$ and applying the association procedure *not only at the end* (step 12.).

Algorithm 6 The Macaulay-like Axis of Evil algorithm.

1: **procedure** AOE2($\mathbf{X}, \mathsf{N}, \mathfrak{T}(\mathbf{X})) \to \mathcal{G}$ ▷ $\mathcal{G}$ contains a factorized minimal Groebner basis of $I(\mathbf{X})$.

Require: the elements N are in increasing order w.r.t lex, $x_1 < ... < x_n$ and they have been computed via the Jumping algorithm, so that also $\mathbf{X}$ is consequently ordered.

Ensure: the Macaulay-like Axis of Evil factorization.

2: **for** $i = 1$ to n **do**
3: $\Xi_i = x_i - a_{1i}$
4: $L_1[i] = 1$
5: $H_1[i] = 1$
6: $\mathsf{G}_1[i] = x_i$
7: **end for**
8: **for** $j = 2$ to S **do**
9: $L_j = L_{j-1}$
10: $H_j = [\underbrace{j, ..., j}_{n \text{ times}}]$
11: $\mathsf{G}_j = \mathsf{Laz}(\mathsf{N}[1, ..., j])$ ▷ Laz is one step of Lazard's algorithm.
12: **if** $\alpha_{jn} = 0$ **then**
13: $R_j = Assoc(\tau_j, \Xi_1, ..., \Xi_n, L_1, ..., L_{j-1}, \mathsf{N}[1, ..., j])$
14: $f_j = \prod_{k=1}^{|R_j|} R_j[k]$
15: $q_j = \frac{f_j}{f_j(P_j)}$
16: **end if**
17: $h = \max(\tau_j)$
18: $BFS(\mathfrak{T}(\mathbf{X}), h, P_j)$
19: Test$(\tau_j, h-1)$ ▷ Test is a procedure which compares the coordinates of the points as explained in step 10 of the algorithm.
20: **if** $\tau_j = x_h^m$ **then**
21: $\Xi_h = x_h - a_{jh}$
22: $L_j[h] = |\Xi_h|$
23: **for** $l = h + 1$ to n **do**
24: $\Xi_l[|\Xi_l|] = Interp(P_j, \Xi_l[|\Xi_l|], \mathsf{N}[1, ..., j], H_j)$
25: **end for**
26: **else**
27: **for** $l = h$ to n **do**
28: $\Xi_l[|\Xi_l|] = Interp(P_j, \Xi_l[|\Xi_l|], \mathsf{N}[1, ..., j], H_j)$
29: **end for**
30: **end if**
31: **end for**
32: **for** $h = 1$ to $|\mathsf{G}_S|$ **do**
33: $\mathcal{G} = Assoc(\sigma_h, \Xi_1, ..., \Xi_n, L_1, ..., L_S, \mathsf{N})$
34: **end for**
35: **end procedure**

Algorithm 7 The Association Procedure.

procedure ASSOC($\sigma, \Xi_1, ..., \Xi_n, L_1, ..., L_i, \mathsf{N} \to R$) ▷ R contains the linear factors.

Require: the elements N are in increasing order w.r.t the lexicographical order w.r.t. $x_1 < ... < x_n$ and they have been computed via the Jumping algorithm, so that also $\mathbf{X}$ is consequently ordered.

Ensure: the factors for the polynomial in the minimal basis whose head is σ.

$h = \max(\sigma)$
$R = [\Xi_h[1], ..., \Xi_h[j_h]]$
$\sigma' = x_h^{j_h}$
$P = \Phi^{-1}_{Jumping}(\sigma')$
while$(\sigma' \neq \sigma)$
$i = \max(\frac{\sigma}{\sigma'})$
$R = R \cup [\Xi_i[L_P[i]]]$
$\sigma' = \sigma' x_i$
$P = \Phi^{-1}_{Jumping}(\sigma)$
end while
end procedure

Algorithm 8 The Interpolation Procedure.

procedure INTERP($P_j, p, \mathsf{N}, H_j, \mathsf{N} \to p$) ▷ p is the factorized polynomial.

Require: the elements N are in increasing order w.r.t the lexicographical order w.r.t. $x_1 < ... < x_n$ and they have been computed via the Jumping algorithm, so that also $\mathbf{X}$ is consequently ordered.

Ensure: interpolation of p in P_j.

$v = p(P_j)$
if $\mathsf{T}(p) \leq \tau_j$ **then**
    $h = \max(\tau_j)$
    $s = H_j[h-1]$
    $p = p - vq_s$
else
    $p = p - vq_j$
end if
end procedure

Algorithm 9 The BFS Procedure.

procedure BFS($\mathfrak{T}(\mathbf{X}), h, P_k$) $\rightarrow L$ ▷ L is the list of factors associated to P_k.

Require: $\mathfrak{T}(\mathbf{X})$ is the children trie, h is the maximal level of $\mathfrak{T}(\mathbf{X})$ we have to deal with and P_k the point under consideration.

Ensure: The BFS of the children trie.

```
    for i = 1 to n do
        for j = 1 to 𝔗(X)[i] do                ▷ 𝔗(X)[i] is the number of nodes at level i
            if 𝔗(X)[i][j][1] == P_k then
                L_k[i] = j
                T = Subtree(𝔗(X)[i][j], 𝔗(X))
                break;
            end if
            if 𝔗(X)[i][j][1][i] == P_k[i] then
                L_k[i] = L_{𝔗(X)[i][j][1]}[i]
                L_k[i] = 𝔗(X)[i][j][1]
                T = Subtree(𝔗(X)[i][j], 𝔗(X))
                break;
            end if
        end for
    end for
end procedure
```

Remark 4.4.4. Let $\{(a_{i,1}, ..., a_{i,l}, *, ..., *)\}$ be the points in biunivocal correspondence to the terms of a certain x_l-range.
By construction, the factors having x_l as head vanish on the point of the shape $(a_{i,1}, ..., a_{i,l}) \in \mathbf{k}^l$ and we have exactly one factor for each x_l-range contained in the first x_{l+1}-range.
By the association procedure, we can observe that, for each h, the first l x_h-factors vanish at the points, corresponding via the Jumping algorithm, to terms τ with $\max(\tau) = x_h$ and $deg_h(\tau) < l$.

Remark 4.4.5. Consider a point $P_j \in \mathbf{X}$, corresponding to a term $\tau_j \in \mathsf{N}(I(\mathbf{X}))$, and the execution of our algorithm on it, referring especially to step 8. For each $m = 1, ..., h-1$, consider the set $\Omega_m = \{P \in \mathbf{X_{j-1}} \,|\, \pi_m(P) = \pi_m(P_j)\}$. In step 8 we are looking for the point $P_l \in \Omega_m$, such that $\tau_l = \min(\Phi_{Jumping}(\Omega_m))$. Performing it, we do not need to scan all the points in $\mathbf{X_{j-1}}$, but only one for each element of the class Σ_m in the children trie. This is a facility provided by the jumping algorithm: the first element of each Σ_m is always put in biunivocal correspondence with the minimal lexicographical term in the class.

Remark 4.4.6. We point out that the computation of the triangular polynomial and the interpolation process come directly from Moeller algorithm (see section 1.6).

The algorithm ends in a finite number of steps, performing loops in a finite set of points and terms.
Suppose now that the test of step 10 passes for each point so that we continue with the algorithm in this chapter until we reach the last point and we prove that our new algorithm is correct. First of all, we need the following

Lemma 4.4.7. Let $\mathsf{N}(I)$, $|\mathsf{N}(I)| < \infty$ be the Groebner escalier of a zerodimensional radical ideal I and let $\mathsf{N}(J) = \mathsf{N}(I) \cup \{\tau\}$, $\tau = \max_{Lex}(\mathsf{N}(J))$. If $x_k > \min(\tau)$, then $x_k\tau \notin \mathsf{G}(J)$.

Proof: By assumption, $\tau \in \mathsf{G}(I)$, $\tau = \max_{Lex}(\mathsf{N}(J))$. Let $x_k > \min(\tau)$.
Since $\frac{x_k\tau}{\min(\tau)} > \tau$, then $\frac{x_k\tau}{\min(\tau)} \notin \mathsf{N}(J)$, so $x_k\tau \notin \mathsf{G}(J)$, by the characterization of $\mathsf{G}(J)$[6]. $\diamond$

Proposition 4.4.8. With the previous notation, $\mathcal{G} := \mathcal{G}_S = \mathcal{G}(I(\mathbf{X}))$.

Proof: It is obvious that $\mathcal{G}_1 = \mathcal{G}(I(\{P_1\}))$. Suppose that $\mathcal{G}_{i-1} = \mathcal{G}(I(\{P_1, ..., P_{i-1}\}))$. We prove that the analogous equality holds for $\mathcal{G}_i$.
First of all, we point out that, since the Groenbner escalier has been constructed via the Jumping algorithm, the term $\sigma_i = x_1^{\beta_1} \cdots x_n^{\beta_n}$, associated to P_i is the maximal term in N_i

[6] At least one of the predecessors of $x_k\tau$ does not belong to $\mathsf{N}(J)$.

w.r.t. lex (by the lex trie construction, described in [37, 67]).
By the above comment and by lemma 4.4.7, if $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{G}_i := \mathsf{G}(I(\{P_1, ..., P_i\}))$ then, either $\tau = x_k\sigma_i$, $x_k \leq \min(\tau)$ or $\tau \in \mathsf{G}_{i-1}$.

In the first case, we observe that $\sigma_i \in \mathsf{G}_{i-1}$, so f_{σ_i} vanishes[7] at $P_1, ..., P_{i-1}$ and $f_{\sigma_i} \mid f_{x_k\sigma_i}$ by the association procedure.
Moreover, the exponents' list of σ_i identifies the first point not annihilating f_{σ_i} (the first x_k-range whose corresponding points do not make f_{σ_i} vanish is the x_k-range containing σ_i).
The interpolation procedure and the association procedure on the variable x_k ensure then that $f_{x_k\sigma_i}$ vanishes at $P_1, ..., P_i$. Indeed f_{σ_i} vanishes in $P_1, ..., P_{i-1}$ and the factor in x_k we take vanishes in P_i.

In the second case, namely $\tau \in \mathsf{G}_{i-1}$, it can be either $\tau > \sigma_i$ or $\tau < \sigma_i$.
In order to continue, we need the technical fact proved below.

Fact 4.4.9. *For the case $\tau > \sigma_i$, only two possibilities may arise, namely:*

A) $\tau = x_h$, $x_h > \max(\sigma_i)$;

B) $\tau = x_{j_0}^{\alpha_{j_0}} x_{j_0+1}^{\alpha_{j_0+1}} \cdots x_n^{\alpha_n}$, *with* $\alpha_n = deg_n(\sigma_i)$, $\alpha_{n-1} = deg_{n-1}(\sigma_i)$,...,$\alpha_{j_0+1} = deg_{j_0+1}(\sigma_i)$ *and* $\alpha_{j_0} = deg_{j_0}(\sigma_i) + 1$.

In order to prove the assertion, we first prove that two variables $x_l > x_k \geq \max(\sigma_i)$ cannot appear with nonzero exponent in τ. Indeed, if it was so, $\frac{\tau}{x_k} \in \mathsf{N}_i$ (being $\tau \in \mathsf{G}_i$) and $\frac{\tau}{x_k} > \sigma_i$, that contradicts the maximality of $\sigma_i \in \mathsf{N}_i$.
On the other hand, if some $x_k \geq \max(\sigma_i)$ appears in τ and $deg_k(\tau) = deg_k(\sigma_i) + l$, with $l > i$, again $\frac{\tau}{x_k} \in \mathsf{N}_i$ and $\frac{\tau}{x_k} > \sigma_i$, thus also this possibility cannot occur.
By the comments above, if $x_h > \max(\sigma_i)$, $x_h \mid \tau$, then any other $x_l \geq \max(\sigma_i)$ does not divide τ and, moreover, $\deg_h(\tau) = 1$.
Being $\sigma_i \nmid \tau$ ($\sigma_i, \tau \in \mathsf{G}_{i-1}$), for $j = 1, ..., \max(\sigma_i)$, it cannot be always $\alpha_j \geq \beta_j$, so $\exists k \in \{1, ..., \max(\sigma_i)\}$ with $\alpha_k < \beta_k$.
If $\alpha_k > 0$, $\frac{\tau}{x_k} \in \mathsf{N}_i$ and $\frac{\tau}{x_k} > \sigma_i$, so this possibility cannot occur.
Otherwise, if $\alpha_k = 0$ and there is some $l \in \{1, ..., \max(\sigma_i)\}$ with $\alpha_l > 0$, by the same argument as before we have a contradiction. Thus, necessarily $deg_h(\tau) = 1$, $\deg_l(\tau) = 0$, for all $l \neq h$.

Let now $\max(\tau) = \max(\sigma_i)$. Then, as $\tau > \sigma$, $\alpha_n > \beta_n$ or $\alpha_n = \beta_n$, ..., $\alpha_{j_0+1} = \beta_{j_0+1}$, $\alpha_{j_0} > \beta_{j_0}$, reasoning as above, $\alpha_{j_0} = \beta_{j_0} + 1$. No variables x_l, $l \in \{1, ..., j_0 - 1\}$ can divide τ by

[7] The polynomial f_{σ_i} is such that $\mathsf{T}(f_{\sigma_i}) = \sigma_i$.

the maximality of σ_i *in* N_i. *Thus we conclude that* $\tau = x_{j_0}^{\alpha_{j_0}} x_{j_0+1}^{\alpha_{j_0+1}} \cdots x_n^{\alpha_n}$, *with* $\alpha_n = deg_n(\sigma_i)$, $\alpha_{n-1} = deg_{n-1}(\sigma_i)$,...,$\alpha_{j_0+1} = deg_{j_0+1}(\sigma_i)$ *and* $\alpha_{j_0} = deg_{j_0}(\sigma_i) + 1$.

Let then $\tau \in \mathsf{G}_{i-1}$, $\tau > \sigma_i$. By the above lemma we know that only cases A), B) can occur, so we study them from an interpolation point of view. We know that f_τ already vanishes at $P_1, ..., P_{i-1}$.

A) f_{x_h} vanishes at P_i: it is a straightforward consequence of the interpolation procedure.

B) f_τ vanishes at P_i: the first x_i-range whose corresponding points do not make f_{σ_i} vanish is the one containing σ_i, so the assertion is again a consequence of the interpolation procedure, applied to the x_i-factor corresponding to that range.

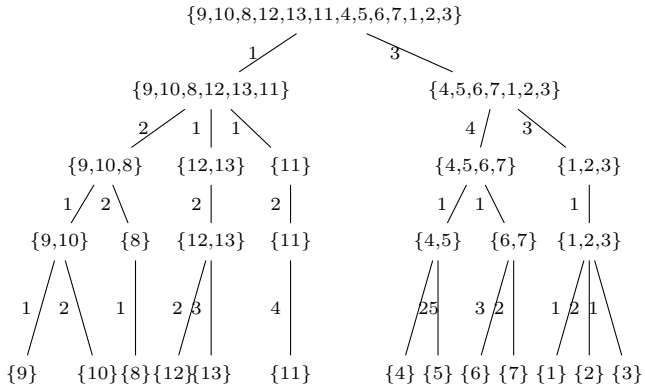
If, instead, $\tau < \sigma_i$, let $x_h = \max(\sigma_i)$. Then, by the correspondence given by the Jumping algorithm, there is a point P_j, sharing with P_i the first $h-1$ coordinates, such that for the corresponding term σ_j $deg_h(\sigma_j) = deg_h(\sigma_i)$. If f_τ vanishes at P_j, then it also vanishes at P_i, by the association procedure. If $\sigma_j < \tau$,f_τ vanishes at P_j and then in P_i (remark 4.4.4). Otherwise, we can we can repeat with σ_j instead of σ_i and conclude by induction. $\diamond$

Remark 4.4.10. The algorithm works correctly in each characteristic for the base field.

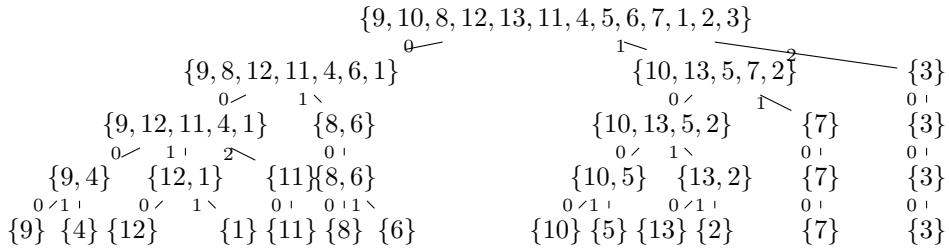
Remark 4.4.11. The same interpolating algorithm can also be used in order to compute an Axis of Evil factorization for the border basis of our ideal $I(\mathbf{X})$.
Once computed the border set $\mathsf{B}(I(\mathbf{X}))$, we proceed as before. The only modification needed is in the association procedure, since, in step (2) it can happen to obtain a term $\tau \notin \mathsf{N}(I(\mathbf{X}))$, so it is impossible to apply $\Phi_{Jumping}$. This is the case for terms $\tau \in \mathsf{B}(I(\mathbf{X})) \setminus \mathsf{G}(I(\mathbf{X}))$. We solve the problem picking randomly the needed number of factors in the lists $\Xi_1, ..., \Xi_n$ involved by the variables of τ not already associated to a factor.
In order to get the border basis from the factorization, clearly, we have to reduce.

We show now some examples of the execution.

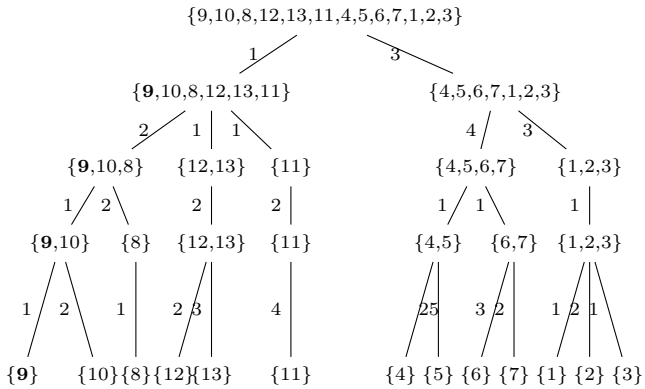
Example 4.4.12. We consider again the set
$\mathbf{X_1} = \{(1,1,2,3), (1,1,2,4), (1,1,2,5), (1,2,1,1), (1,2,1,2), (1,2,2,1), (1,2,2,2),$
$(3,1,1,2), (3,1,2,2), (3,1,2,3), (3,3,1,1), (3,4,1,1), (3,4,1,2)\} = \{P_1, ..., P_{13}\}$.
We have reordered the points via the jumping algorithm in examples 4.3.1, 4.3.4 and 4.3.5, so we have already constructed both the children trie and the lex trie, i.e.



and

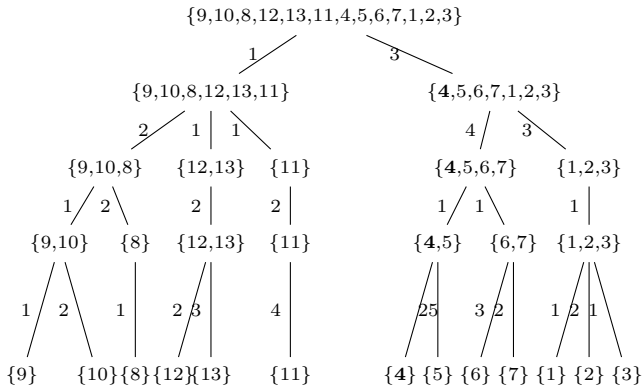


Therefore, we can start the new interpolation process.
We denote in boldface the points involved by the BFS. The first point is $P_9 = (3, 1, 2, 2)$, corresponding $\tau_9 = 1$. The linear factors involved here are trivially $X = \{x - 3\} = \{\mathsf{x}_1\}$, $Y = \{y - 1\} = \{\mathsf{y}_1\}$, $Z = \{z - 2\} = \{\mathsf{z}_1\}$, $T = \{t - 2\} = \{\mathsf{t}_1\}$, while $L_9 = [1, 1, 1, 1]$.

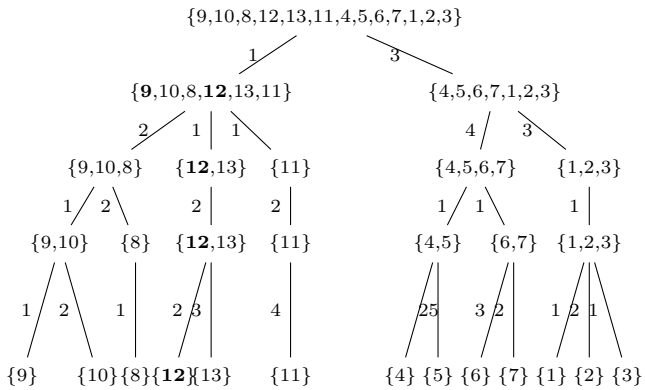


The second point in the new configuration is $P_4 = (1, 2, 1, 1)$, corresponding to $\tau_4 = x$, while the triangular polynomial is $q_4 = \frac{1}{ev_{P_4}(x_1)} = \frac{-1}{2}(x - 3)$. The minimal monomial basis

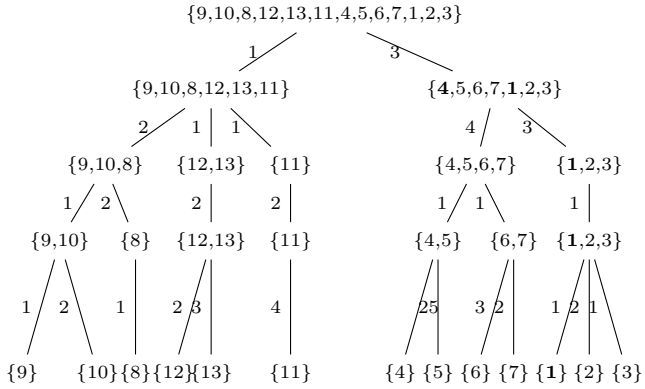
is $\mathsf{G} = \{x^2, y, z, t\}$.
We list now the factors:
$X = \{x-3, x-1\} = \{\mathsf{x}_1, \mathsf{x}_2\}$: we add a new factor in x.
$Y = \{y + \frac{1}{2}x - \frac{5}{2}\} = \{\mathsf{y}_1\}$: we assign to y_1 the new value $\mathsf{y}_1 - ev_{P_4}(\mathsf{y}_1)q_4$.
$Z = \{z - \frac{1}{2}x - \frac{1}{2}\} = \{\mathsf{z}_1\}$: $\mathsf{z}_1 \to ev_{P_4}(\mathsf{z}_1)q_4$.
$T = \{t - \frac{1}{2}x - \frac{1}{2}\} = \{\mathsf{t}_1\}$: $\mathsf{t}_1 \to ev_{P_4}(\mathsf{t}_1)q_4$.
We have $L_4 = [2, 1, 1, 1]$:



For $P_{12} = (3, 4, 1, 1)$ we perform as before:
$\mathsf{N} = \{1, x, y\}$, $q_{12} = \frac{1}{ev_{P_{12}}(\mathsf{y}_1)} = \frac{1}{3}y_1$ and the minimal monomial basis is $\mathsf{G} = \{x^2, xy, y^2, z, t\}$.
The factors are:
$X = \{x-3, x-1\} = \{\mathsf{x}_1, \mathsf{x}_2\}$: the factors in x remain unchanged from now on, so we stop listing them.
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - 4\} = \{\mathsf{y}_1, \mathsf{y}_2\}$: we add y_2, so y_1 remains unchanged from now on.
$Z = \{z + \frac{1}{3}y - \frac{1}{3}x - \frac{4}{3}\} = \{\mathsf{z}_1\}$: $\mathsf{z}_1 \to \mathsf{z}_1 - ev_{P_{12}}(\mathsf{z}_1)q_{12}$.
$T = \{t + \frac{1}{3}y - \frac{1}{3}x - \frac{4}{3}\} = \{\mathsf{t}_1\}$: $\mathsf{t}_1 \to \mathsf{t}_1 - ev_{P_{12}}(\mathsf{t}_1)q_{12}$.
$L_{12} = [1, 2, 1, 1]$:

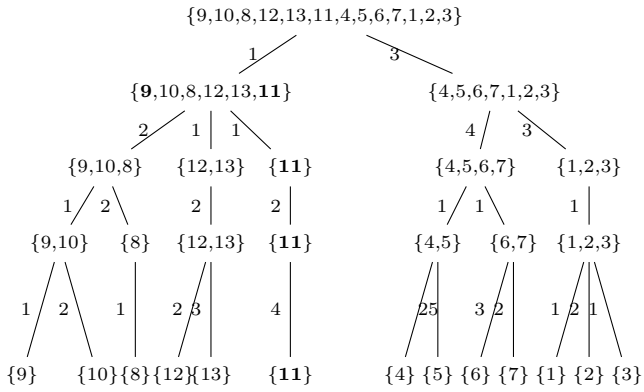


y-factor and then x_1 since P_{12} has 3 as first coordinate. For $P_1 = (1,1,2,3)$ we have:
$\mathsf{N} = \{1, x, y, xy\}$, $q_1 = \frac{1}{ev_{P_1}(x_1 y_1)} = \frac{1}{2}x_1 y_1$ and $\mathsf{G} = \{x^2, y^2, z, t\}$.
For the listed factors we have:
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}\} = \{y_1, y_2\}$: we interpolate y_2, but we cannot use the triangular polynomial q_1 since $T(q_1) = xy > y$. So we go down and pick q_4, obtaining $y_2 \to y_2 - ev_{P_1}(y_2)q_4$.
$Z = \{z - \frac{1}{3}xy + \frac{4}{3}y - \frac{1}{6}x^2 + x - \frac{23}{6}\} = \{z_1\}$: here we can use q_1: $z_1 \to z_1 - ev_{P_1}(z_1)q_1$.
$T = \{t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{91}{12}\} = \{\mathsf{t}_1\}$: $\mathsf{t}_1 \to \mathsf{t}_1 - ev_{P_1}(\mathsf{t}_1)q_1$.
$L_1 = [2, 2, 1, 1]$:

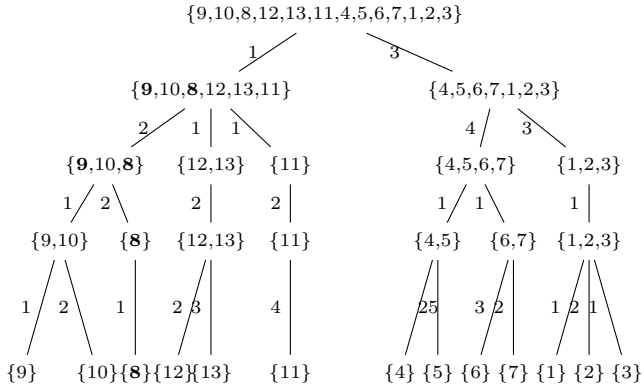


For $P_{11} = (3,3,1,1)$, since $\mathsf{N} = \{1, x, y, xy, y^2\}$ and $q_{11} = \frac{-1}{2}\mathsf{y}_1\mathsf{y}_2$, we have to add a factor in y and interpolate $\mathsf{z}_1, \mathsf{t}_1$ using q_{11}. The monomial basis is $\mathsf{G} = \{x^2, xy^2, y^3, z, t\}$, whereas the factors are:
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{y_1, y_2, y_3\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}\} = \{\mathsf{z}_1\}$

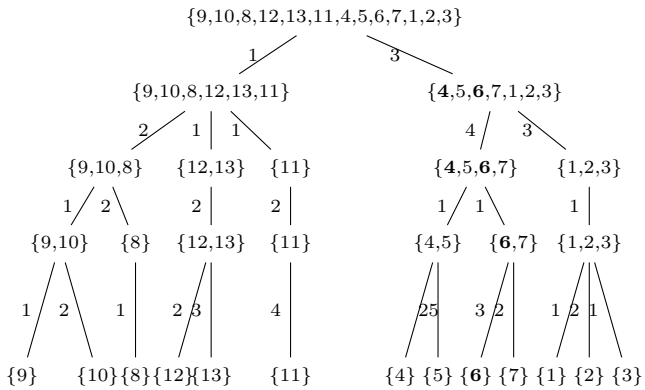
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}\} = \{\mathsf{t}_1\}$.
$L_{11} = [1, 3, 1, 1]$:



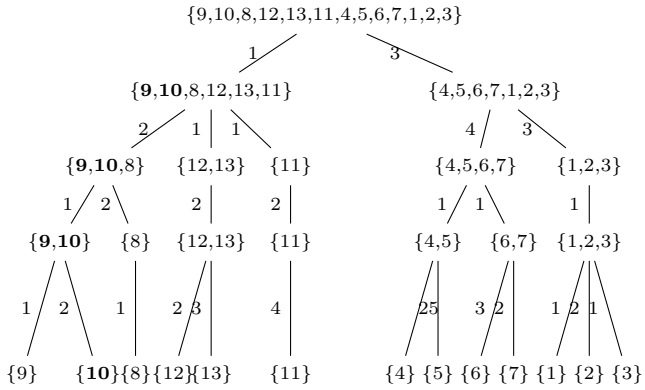
The point $P_8 = (3, 1, 1, 2)$ gives $\mathsf{N} = \{1, x, y, xy, y^2, z\}$, $q_8 = -z_1$ and $\mathsf{G} = \{x^2, xy^2, y^3, xz, yz, z^2, t\}$.
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{\mathsf{y}_1, \mathsf{y}_2, \mathsf{y}_3\}$: from now on, the factors in y remain unchanged, so we stop listing them.
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z - 1\} = \{\mathsf{z}_1, \mathsf{z}_2\}$: we add a new factor.
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}\} = \{\mathsf{t}_1\}$: $\mathsf{t}_1 \to \mathsf{t}_1 - ev_{P_8}(\mathsf{t}_1)q_8$.
$\mathcal{G} = \{x_1x_2, y_1y_2x_1, y_1y_2y_3, x_1z_1, y_1z_1, z_1z_2, t_1\}$. $L_8 = [1, 1, 2, 1]$:



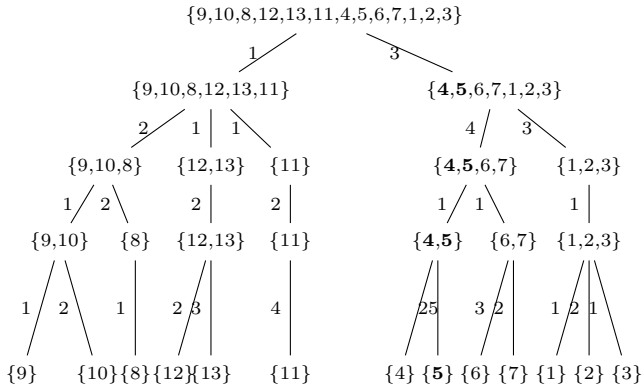
$P_6 = (1, 2, 2, 1)$, $\mathsf{N} = \{1, x, y, xy, y^2, z, xz\}$, $q_6 = -\frac{1}{2}\mathsf{x}_1\mathsf{z}_1$, $\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, t\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{\mathsf{z}_1, \mathsf{z}_2\}$: $z_2 \to z_2 - ev_{P_6}(z_2)q_4$, since $T(q_6) = xz > z$.
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}\} = \{t_1\}$: $\mathsf{t}_1 \to \mathsf{t}_1 - ev_{P_6}(\mathsf{t}_1)q_6$.
$L_6 = [2, 1, 2, 1]$:



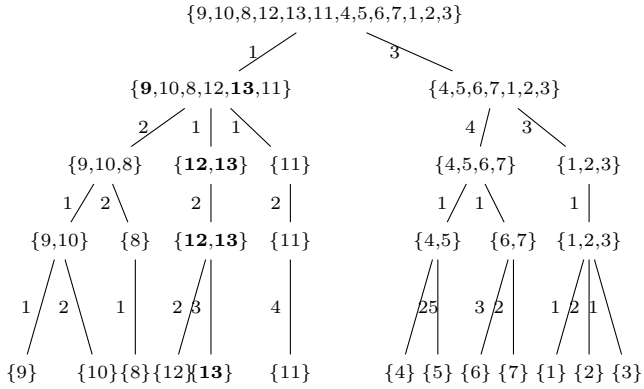
Take now $P_{10} = (3, 1, 2, 3)$, obtaining $\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t\}$. From now on we do not need to compute triangular polynomials anymore.
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, xt, yt, zt, t^2\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{\mathsf{z}_1, \mathsf{z}_2\}$
Since also the factors in z remain unchanged from now on, we stop listing them. $T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t - 3\} = \{\mathsf{t}_1, \mathsf{t}_2\}$: we add a new factor t_2 (t_1 remains always unchanged) and we have $L_{10} = [1, 1, 1, 2]$



For $P_5 = (1, 2, 1, 2)$, we have $\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t, xt\}$ and
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, yt, zt, t^2\}$.
We only have to interpolate t_2, using $\mathsf{t}_2 - ev_{P_5}(\mathsf{t}_2)q_4$:
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t - \frac{1}{2}x - \frac{3}{2}\} = \{\mathsf{t}_1, \mathsf{t}_2\}$
$L_5 = [2, 1, 1, 2]$.

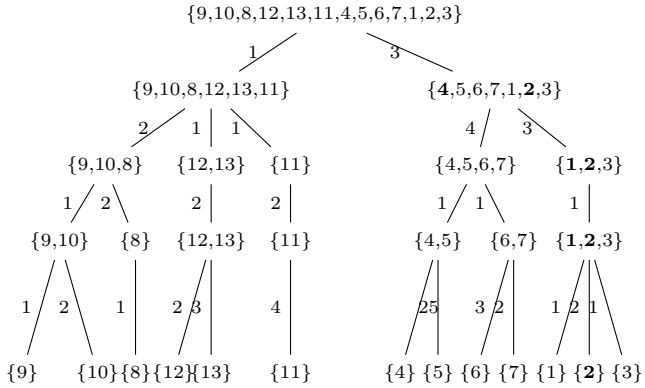


For $P_{13} = (3, 4, 1, 2)$, we have $\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t, xt, yt\}$ and
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, xyt, y^2, zt, t^2\}$.
$X = \{x - 3, x - 1\} = \{x_1, x_2\}$
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{y_1, y_2, y_3\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{z_1, z_2\}$
We interpolate again t_2, but using q_{12}:
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t + \frac{1}{3}y - \frac{1}{3}x - \frac{7}{3}\} = \{\mathsf{t}_1, \mathsf{t}_2\}$, so $t_2 \to \mathsf{t}_2 - ev_{P_{13}}(\mathsf{t}_2)q_{12}$, obtaining
$L_{13} = [1, 2, 1, 2]$.

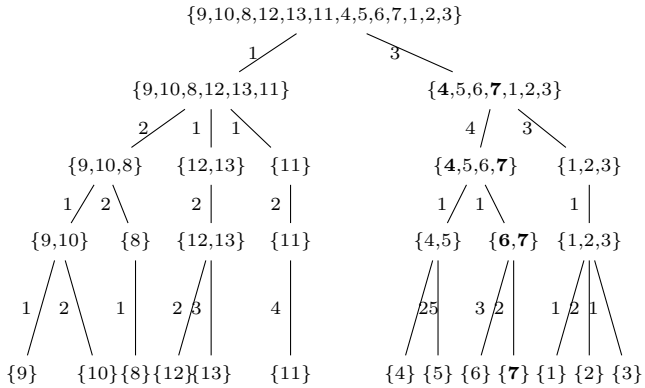


For $P_2 = (1, 1, 2, 4)$, we have
$\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t, xt, yt, xyt\}$ and
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, y^2t, zt, t^2\}$.
$X = \{x - 3, x - 1\} = \{x_1, x_2\}$
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{y_1, y_2, y_3\}$

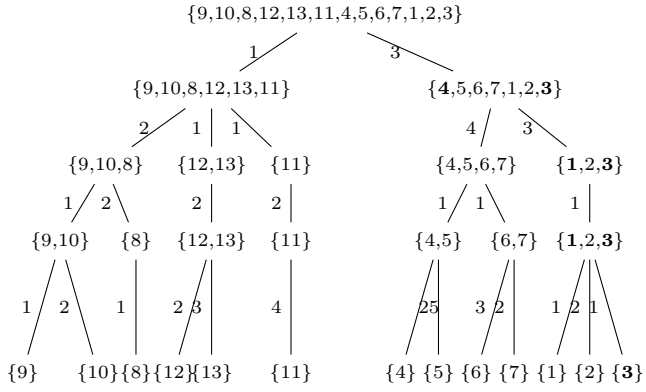
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{z_1, z_2\}$
We interpolate t_2: $\mathsf{t}_2 \to \mathsf{t}_2 - ev_{P_2}(\mathsf{t}_2)q_1$:
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12}\} = \{\mathsf{t}_1, \mathsf{t}_2\}$ and we get
$L_2 = [2, 2, 1, 2]$



The point $P_7 = (1, 2, 2, 2)$ gives
$\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t, xt, yt, xyt, zt\}$,
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, y^2t, xzt, t^2\}$.
$X = \{x - 3, x - 1\} = \{x_1, x_2\}$
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{y_1, y_2, y_3\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{z_1, z_2\}$
Since t_2 vanishes in P_7 there are no changes to perform on the factors:
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12}\} = \{\mathsf{t}_1, \mathsf{t}_2\}$ and we obtain
$L_7 = [2, 1, 2, 2]$.



For $P_3 = (1, 1, 2, 5)$ we have: $\mathsf{N} = \{1, x, y, xy, y^2, z, xz, t, xt, yt, xyt, zt, t^2\}$,
$\mathsf{G} = \{x^2, xy^2, y^3, yz, z^2, y^2t, xzt, xt^2, yt^2, zt^2t^3\}$.
$X = \{x-3, x-1\} = \{x_1, x_2\}$
$Y = \{y + \frac{1}{2}x - \frac{5}{2}, y - \frac{3}{2}x + \frac{1}{2}, y - 3\} = \{y_1, y_2, y_3\}$
$Z = \{z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8}, z + \frac{1}{2}x - \frac{5}{2}\} = \{z_1, z_2\}$
Here we only add a factor:
$T = \{t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8}, t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12}, t - 5\} = \{\mathsf{t}_1, \mathsf{t}_2, \mathsf{t}_3\}$.
$L_3 = [2, 1, 1, 3]$:



The final Groebner basis is:
$\mathcal{G} = \{\mathsf{x}_1\mathsf{x}_2, \mathsf{y}_1\mathsf{y}_2\mathsf{x}_1, \mathsf{y}_1\mathsf{y}_2\mathsf{y}_3, \mathsf{y}_1\mathsf{z}_1, \mathsf{z}_1\mathsf{z}_2, \mathsf{y}_2\mathsf{y}_1\mathsf{t}_1, \mathsf{x}_2\mathsf{z}_1\mathsf{t}_1, \mathsf{t}_1\mathsf{t}_2\mathsf{x}_2, \mathsf{t}_1\mathsf{t}_2\mathsf{y}_2, \mathsf{t}_1\mathsf{t}_2\mathsf{z}_1\mathsf{t}_1\mathsf{t}_2\mathsf{t}_3\}$, i.e.

- $f_1 = (x-3)(x-1)$;
- $f_2 = (y + \frac{1}{2}x - \frac{5}{2})(y - \frac{3}{2}x + \frac{1}{2})(x-3)$;
- $f_3 = (y + \frac{1}{2}x - \frac{5}{2})(y - \frac{3}{2}x + \frac{1}{2})(y-3)$;
- $f_4 = (y + \frac{1}{2}x - \frac{5}{2})(z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8})$;
- $f_5 = (z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8})(z + \frac{1}{2}x - \frac{5}{2})$;
- $f_6 = (y + \frac{1}{2}x - \frac{5}{2})(y - \frac{3}{2}x + \frac{1}{2})(t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})$;
- $f_7 = (x-1)(z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8})(t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})$;
- $f_8 = (x-1)(t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})(t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12})$;
- $f_9 = (t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})(t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12})(y - \frac{3}{2}x + \frac{1}{2})$;

- $f_{10} = (t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})(t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12})(z - \frac{1}{6}y^2 - \frac{1}{6}xy + \frac{5}{3}y - \frac{1}{24}x^2 + \frac{1}{3}x - \frac{29}{8})$;
- $f_{11} = (t - \frac{1}{6}y^2 - \frac{2}{3}xy + \frac{19}{6}y - \frac{7}{24}x^2 + \frac{7}{3}x - \frac{59}{8})(t - \frac{5}{6}xy + \frac{17}{6}y - \frac{5}{12}x^2 + 3x - \frac{103}{12})(t - 5)$;

In this example we have considered a mixed tower structure but we could go to the end of the algorithm.

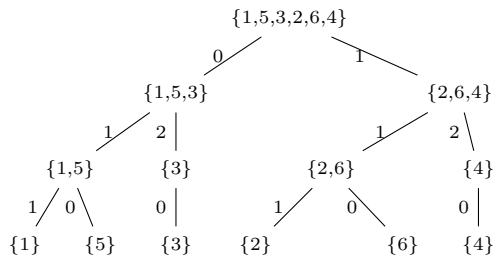
If for some set of points $\mathbf{X}$ we know (for example for theoretical reasons) that it is possible to associate to it an unmixed tower structure, we can appreciably simplify the execution. Indeed, we know that.

- We do not need the sort of BFS on the trie (and actually we do not need the lists H_i, L_i): by the unmixed tower structure we already know that if P_τ corresponds to $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ then $P_{\tau'}$, corresponding to $\tau' = \frac{\tau}{x_i^{\alpha_i} \cdots x_n^{\alpha_n}}$ share its first $(i-1)$ coordinates with P_τ.
- We do not need the association procedure : the factorization we obtain is properly à la Macaulay.
- We do not need to perform any test on the ranges.

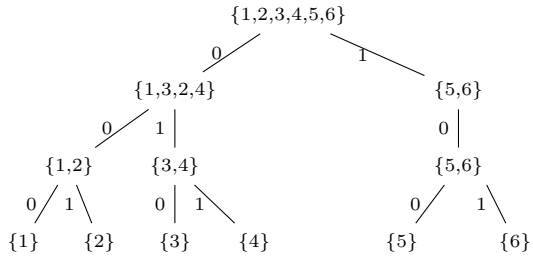
Let us see an example of this situation.

Example 4.4.13. Let us consider the set $\mathbf{X} = \{(0,1,1), (1,1,1), (0,2,0), (1,2,0), (0,1,0), (1,1,0)\}$. Let us start computing $\mathsf{N}(I(\mathbf{X}))$.
We have:
$\Sigma_0 = \{\{1,5,3,2,6,4\}\}$
$\Sigma_1 = \{\{1,5,3\},\{2,6,4\}\}$
$\Sigma_2 = \{\{1,5\},\{3\},\{2,6\},\{4\}\}$
$\Sigma_3 = \{\{1\},\{5\},\{3\},\{2\},\{6\},\{4\}\}$

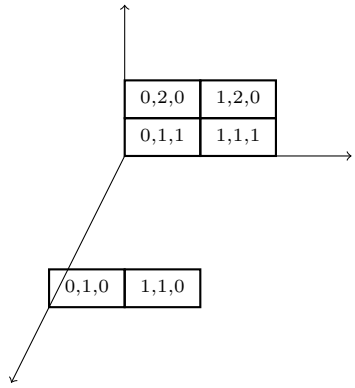
The children trie is



Now we construct the lex trie. For $h = 1$, we have $n - h = 2$, so we iterate on Σ_2, getting
$v_0 = \{1, 3, 2, 4\}$
$v_1 = \{5, 6\}$.
Then we continue with $h = 2$ and, since $n - h = 1$, we iterate on Σ_1, obtaining
$v_{0,0} = \{1, 2\} =: v_0$
$v_{0,1} = \{3, 4\} =: v_1$
$v_{1,0} = \{5, 6\} =: v_2$
Finally, for $h = 3$, we iterate on Σ_0 and we finally get $v_{0,0} = \{1\}$
$v_{0,1} = \{2\}$
$v_{1,0} = \{3\}$
$v_{1,1} = \{4\}$
$v_{2,0} = \{5\}$
$v_{2,1} = \{6\}$
and the lex trie is



Now, we deal with the factorization, iterating on the points. In this case, the tower structure is unmixed, so we can simplify the execution:



Let us start with $P_1 = (0, 1, 1)$, corresponding to $\tau_1 = 1 \in \mathsf{N}(I)$. The associated triangular polynomial is $q_1 = 1$ and, up to now, the linear factors are

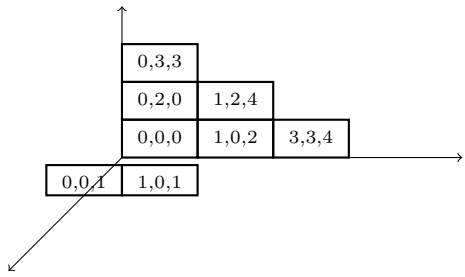
$\Xi_1 = \{x_1\}$
$\Xi_2 = \{x_2 - 1\}$
$\Xi_3 = \{x_3 - 1\}$.
The second point, $P_2 = (1, 1, 1)$, corresponds to $\tau_2 = x_1$ and we have $q_2 = x_1$. The lists of factors are
$\Xi_1 = \{x_1, x_1 - 1\}$: we added a new polynomial in x_1
$\Xi_2 = \{x_2 - 1\}$
$\Xi_3 = \{x_3 - 1\}$.
Consider now $P_3 = (0, 2, 0)$, corresponding to $\tau_3 = x_2$ and to the triangular polynomial $q_3 = x_2 - 1$.
The lists of factors are
$\Xi_1 = \{x_1, x_1 - 1\}$
$\Xi_2 = \{x_2 - 1, x_2 - 2\}$: we added a new polynomial in x_2
$\Xi_3 = \{x_3 + x_2 - 2\}$: we have interpolated as $x_3 - 1 \to (x_3 - 1) + q_3$.
For $P_4 = (1, 2, 0)$ we have $\tau_4 = xy$, $q_4 = x_1(x_2 - 1)$ and the factors become:
$\Xi_1 = \{x_1, x_1 - 1\}$
$\Xi_2 = \{x_2 - 1, x_2 - 2\}$
$\Xi_3 = \{x_3 + x_2 - 2\}$.
For $P_5 = (0, 1, 0)$, we get $\tau_5 = x_3$ and we do not compute the triangular polynomial, since its head term would be x_3 and we cannot use it to interpolate.
The list of factors are
$\Xi_1 = \{x_1, x_1 - 1\}$
$\Xi_2 = \{x_2 - 1, x_2 - 2\}$
$\Xi_3 = \{x_3 + x_2 - 2, x_3\}$: we added a new factor in x_3.
For $P_6 = (1, 1, 0)$, we get $\tau_6 = x_1x_3$ and, as for P_5, we have no need to compute the triangular polynomial.
The final list of factors are
$\Xi_1 = \{x_1, x_1 - 1\}$
$\Xi_2 = \{x_2 - 1, x_2 - 2\}$
$\Xi_3 = \{x_3 + x_2 - 2, x_3\}$.
The factorization we get is:

$$\mathcal{G} = \{x_1(x_1 - 1), (x_2 - 1)(x_2 - 2), (x_3 + x_2 - 2)(x_2 - 1), x_3(x_3 + x_2 - 1)\}$$

and it is an Axis of Evil factorization à la Macaulay

The version of the Axis of Evil algorithm we are examining now displays many differences with the original one.
The factors are updated at each step and not computed each time from the beginning. Moreover, some linear factors are used several times i.e. in relation with more than one head, even if they have been computed only once.
This was not in the original Axis of Evil procedure, where *for each term* in $\mathsf{G}(I)$ it was necessary to interpolate specifically each factor, possibly computing the same factor more than once.

Example 4.4.14. Given $\mathbf{X} = \{(0,0,0),(1,0,2),(3,3,4),(0,2,0),(1,2,4),(0,3,3),(0,0,1),(1,0,1)\}$, we perform on it both the original Axis of Evil algorithm and the second version.
We suppose $\mathbf{X}$ and $\mathsf{N}(I(\mathbf{X})) = \{1, x, x^2, y, xy, y^2, z, xz\}$ be ordered as provided by the jumping algorithm. As explained before, this is *necessary* only for the second version.
The tower structure turns out to be



The monomial basis is $\mathsf{G}(I(\mathbf{X})) = \{x^3, x^2y, xy^2, y^3, x^2z, yz, z^2\}$.
The original algorithm produces:

- $f_1 = x(x-1)(x-3)$;
- $f_2 = x(x-1)(y-3)$;
- $f_3 = x(y-2)(y-\frac{3}{2}x+\frac{3}{2})$;
- $f_4 = y(y-2)(y-3)$;
- $f_5 = x(x-1)(z-4)$;
- $f_6 = y(z-3y+\frac{11}{6}x^2-\frac{35}{6}x+6)$;
- $f_7 = (z-1)(z-y^2-xy+2y+\frac{7}{3}x^2-\frac{13}{3}x)$.

All the repeated factors have been computed each time they appear in the factorization, so, for example, we compute the same factor $(x-1)$ three times.
Consider now the second algorithm. The factorized basis we get is

- $x(x-1)(x-3)$;
- $x(x-1)(y-\frac{1}{2}x(x-1))$;
- $x(y-\frac{1}{2}x(x-1))(y-2)$;
- $(y-3)(y-\frac{1}{2}x(x-1))(y-2)$;
- $x(x-1)(z-y^2+\frac{1}{2}x^2y-\frac{3}{2}xy+2y+\frac{1}{2}x^3-\frac{7}{6}x^2-\frac{4}{3}x)$;
- $(y-\frac{1}{2}x(x-1))(z-y^2+\frac{1}{2}x^2y-\frac{3}{2}xy+2y+\frac{1}{2}x^3-\frac{7}{6}x^2-\frac{4}{3}x)$;
- $(z-y^2+\frac{1}{2}x^2y-\frac{3}{2}xy+2y+\frac{1}{2}x^3-\frac{7}{6}x^2-\frac{4}{3}x)(z-1)$.

In this case, even if a factor repeats more than once in the factorized basis, it is computed *only once*.

There is something more: in this new version of the Axis of Evil algorithm, we interpolate at each point $P \in \mathbf{X}$, only in some variables.
More precisely, if $\Phi_{Jumping}(P) = \tau$ and $\max(\tau) = x_h$, we interpolate only in $x_h, ..., x_n$.
In the original algorithm, we compute separately all the needed factors.
This means that the number of computed factors decreases with the second version of the algorithm.
In 4.4.14, for example, we noticed that, from P_4, the list of the factors in x maintains unchanged. The same happens for the y factors from P_7 on.

Let q_i be the triangular polynomial associated to a point $P_i \in \mathbf{X}$.
We have $\mathsf{T}(q_i) = \Phi_{Jumping}(P_i) = \tau_i$. If τ_i is bigger than the variable in which we are interpolating it is not possible to use q_i because if we do it, we would change the leading term of the linear factor.
For example, if $\Phi_{Jumping}(P_i) = xy$ we cannot interpolate the y factor vanishing in P_i using q_i.
We would need then another triangular polynomial, but *we don't have to compute it*, thanks to the list L, constructed exploiting the sort of BFS we perform on the children tree 4.4.1.

We also notice that, when we reach a term $\tau \in \mathsf{N}(I(\mathbf{X}))$ such that $\max(\tau) = x_n$ we do not need to compute any triangular polynomial more: the ones we have are enough in order to perform the whole interpolation step.

Example 4.4.15. If we take, for example, the set
$\mathbf{X} = \{(1,1),(2,3),(1,2),(2,4),(1,3),(2,6),(1,4),(2,5),(1,6),(2,7),(1,11),(2,14)\} \subset \mathbf{k}^2$, the Groebner escalier is
$\mathsf{N}(I(\mathbf{X})) = \{1, x, y, xy, y^2, xy^2, y^3, xy^3, y^4, xy^4, y^5, xy^5\} \subset \mathbf{k}[x,y]$.
While performing our algorithm, we only have to compute and store the triangular polynomial associated to $(2,3)$, even if $|\mathbf{X}| = 12$. This happens because the term corresponding to the third point, i.e. $(1,2)$, contains the maximal variable and so does every subsequent term.

The arrangement of $\mathbf{X}$ in towers is all we need in order to interpolate: once it is given, we exactly know which are the points and the polynomials to pick in order to obtain the correct factors.
In the original Axis of Evil we had to check at each step which points already vanish in a partial factorized polynomial.
On the other hand, computational evidence shows that in general the linear factors obtained from the original Axis of Evil algorithm are sparser than the ones obtained via the new algorithm. Actually, for the first version, we know that the number of terms for a linear factor is bounded above by $|\mathbf{X}| + 1$ (the leading term plus as many terms as the points to interpolate in, by the interpolation step of 5), whereas we do not have such a bound for the second version. Moreover, in the second version, the factors are not reduced.

Let us deal with Macaulay's trick and the Axis of Evil.

Example 4.4.16. Taken the set $\mathbf{X} = \{(0,0),(1,2),(0,3)\}$, we have $\mathsf{N}(I(\mathbf{X})) = \{1, x, y\}$ and $\mathsf{G}(I(\mathbf{X})) = x^2, xy, y^2$.
The first Axis of Evil factorization is

- $f_1 = x(x-1)$;
- $f_2 = x(y-2)$;
- $f_3 = (y-3)(y-2x)$.

while the second one is

- $f_1 = x(x-1)$;
- $f_2 = x(y-2x)$;
- $f_3 = (y-2x)(y-3)$.

A factorization à la Macaulay requires only two factors in x and two factors in y, so the first factorization is not à la Macaulay, while the second does.

Remark 4.4.17. Given a finite set of distinct points $\mathbf{X}$, the Axis-of-Evil theorem finds for the lexicographical Groebner basis of $I(\mathbf{X})$ a factorization linear in the leading terms, passing through the lexicographical Groebner escalier à la Cerlienco-Mureddu $\mathsf{N}(\mathbf{X})$ of $I(\mathbf{X})$, while Macaulay's trick, given an order ideal N finds a set of points $\widetilde{\mathbf{X}}$ such that $\mathsf{N}(I(\widetilde{\mathbf{X}})) = N$ and the lexicographical Groebner basis of $I(\widetilde{\mathbf{X}})$ is linearly factorized .
If $\mathbf{X}$ is a finite set of distinct points as generated by Macaulay's trick, the Axis-of-Evil factorization is linear, not only in the heads.
If $\mathbf{X}$ is a finite set of arbitrary distinct points, the Axis-of Evil factorization is not really linear and, given an order ideal N there exist sets $\mathbf{X}$ of distinct points such that $\mathsf{N}(I(\mathbf{X})) = \mathsf{N}$, but the lexicographical Groebner basis of $I(\mathbf{X})$ has no linear factorization à la Macaulay.
We display now an example which shows that *the Axis of Evil algorithm makes Macaulay's trick not work.*
Consider again the set $\mathbf{X_0} = \{(3,0,0),(3,1,4),(1,2,3),(1,2,5)\}$ and the polynomial ring $\mathbf{k}[x,y,z]$ equipped with the lexicographical order induced by $1 < x < y < z$. The Groebner escalier associated to $I(\mathbf{X_0})$ is $\mathsf{N}(I(\mathbf{X_0})) = \{1, x, y, z\}$, while the minimal monomial basis of the initial ideal is $\mathsf{G}(I(\mathbf{X_0})) = \{x^2, xy, xz, y^2, yz, z^2\}$.
According the second procedure, there should be two factors whose leading term is x, say X_1, X_2, two factors whose leading term is y (Y_1, Y_2) and two factors whose leading term is z (Z_1, Z_2). These factors should be of the following form:

- $x + a$, $a \in k$;
- $y + f(x)$, $f(x) \in \mathbf{k}[x]$;
- $z + g(x,y)$, $g(x,y) \in \mathbf{k}[x,y]$.

Focus on xy, xz. If Macaulay's trick holds in the required Groebner basis there should be both X_1Y_1 and X_1Z_1. The factor X_1 can be only $(x-1), (x-3)$, so there are two cases:

- $X_1 = (x-1)$: the polynomial $(x-1)Y_1$ must vanish on all the points of $\mathbf{X_0}$. We know that it vanishes on $(1,2,3),(1,2,5)$ because of the factor $(x-1)$, so Y_1 should vanish on $(3,0,0),(3,1,4)$. It means that it must hold simultaneously $f(3) = 0$ and $f(3) = -1$, but the evaluation of a polynomial $f(x) \in \mathbf{k}[x]$ is unique, so we have a contradiction;
- $X_1 = (x-3)$: there are no problems for $(x-3)Y_1$, while we encounter an analogous contradiction for $(x-3)Z_1$. The latter should vanish on all the points of $\mathbf{X_0}$ and we know that it does for $(3,0,0),(3,1,4)$. This means that Z_1 should vanish on both

$(1,2,3)$ and $(1,2,5)$, so it should hold $g(1,2) = -3$ and $g(1,2) = -5$, i.e. again a contradiction.

This example is minimal, since if we remove a point from $\mathbf{X_0}$ the argument does not work anymore.

Notice that the problem is related to the left shifting of the towers we have in the second z-range.

Conversely, we can show that the Axis-of-Evil context includes cases which are not contemplated by Macaulay's trick.

Take for example $\mathsf{N} = \{1, x, y, z\} \subseteq \mathbf{k}[x, y, z]$, imposing, as usual, the lexicographical order with $x < y < z$. Macaulay recovers from N a set of points $\mathbf{X}$ and a set $\mathcal{G}$ of polynomials such that, called $I = I(\mathbf{X})$, $\mathsf{N} = \mathsf{N}(I)$ and $\mathcal{G}$ is the reduced Groebner basis of I.

We stress the fact that Macaulay's trick imposes strong conditions on the set of points, so it does not recover all the sets of points with a given Groebner escalier N.

First of all, Macaulay recovers from $\mathsf{N}(I)$ the monomial basis $\mathsf{G}(I)$. In our example $\mathsf{G}(I) = \{x^2, xy, xz, y^2, yz, z^2\}$.

In $\mathsf{G}(I)$ he isolates the pure powers of all the variables, which are present there since N is a finite set: $d_1 = d_x = 2$, $d_2 = d_y = 3$, $d_3 = d_z = 2$. After that, for each i, j, l, $j \neq l$ takes the elements

$$a_{i,j} \in k,\ i = 1, ..., 3,\ j = 1, ..., d_i,\ a_{i,j} \neq a_{i,l},$$

namely $a_{1,0} = 1$, $a_{1,1} = 2$, $a_{2,0} = 3$, $a_{2,1} = 4$, $a_{3,0} = 5$, $a_{3,1} = 6$.

The polynomials in the reduced Groebner basis are defined by the following formula, where $\mathfrak{X} = [x, y, z]$:

$$g_m = \prod_{i=1}^{3} \prod_{j=0}^{e_{i,m}-1} (\mathfrak{X}[i] - a_{i,j}),\ m \in \mathsf{G}(I).$$

In our example we have:

$g_1 = (x-1)(x-2)$, $T(g_1) = x^2$

$g_2 = (x-1)(y-3)$, $T(g_2) = xy$

$g_3 = (x-1)(z-5)$, $T(g_3) = xz$

$g_4 = (y-3)(y-4)$, $T(g_4) = y^2$

$g_5 = (y-3)(z-5)$, $T(g_5) = yz$

$g_6 = (z-5)(z-6)$, $T(g_6) = z^2$.

Finally $\mathbf{X} = \{(a_{1,e_1}, ..., a_{3,e_3}) \in \mathbf{k}^n \,|\, x^{e_1} y^{e_2} z^{e_3} \in \mathsf{N}\}$, i.e.

$$\mathbf{X} = \{(1,3,5), (2,3,5), (1,4,5), (1,3,6)\}.$$

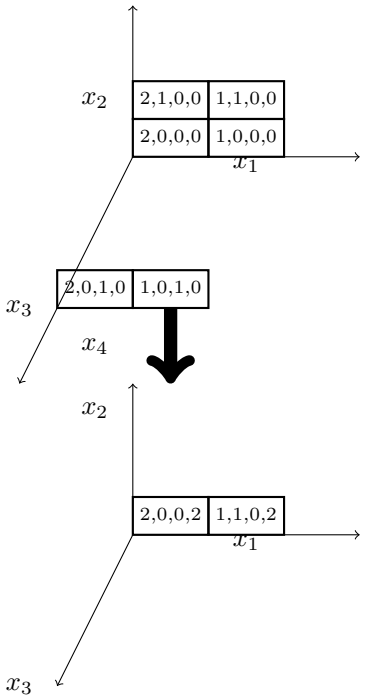
For example, also $\mathbf{X}' = \{(1,3,5), (2,3,5), (1,4,\mathbf{7}), (1,3,6)\}$ has the same Groebner escalier as $\mathbf{X}$, but we cannot recover it since there are only two possible third coordinates.

We display now an example of tower structure making our algorithm stop before getting the whole factorization.

Example 4.4.18. Let us consider the set

$$\mathbf{X} = \{(2,0,0,0), (1,0,0,0), (2,1,0,0), (1,1,0,0), (2,0,1,0), (1,0,1,0), (2,0,0,2), (1,1,0,2)\}$$

with tower structure



The Groebner escalier is $\mathsf{N}(I(\mathbf{X})) = \{1, x_1, x_2, x_1x_2, x_3, x_1x_3, x_4, x_1x_4\}$ and the monomial basis is $\mathsf{G}(I(\mathbf{X})) = \{x_1^2.x_2^2, x_2x_3, x_3^2, x_2x_4, x_3x_4, x_4^2\}$. Let us start with $P_1 = (2,0,0,0)$, which corresponds to $\tau_1 = 1$ and to the triangular polynomial $q_1 = 1$. The lists of factors are
$\Xi_1 = \{x_1 - 2\}$
$\Xi_2 = \{x_2\}$
$\Xi_3 = \{x_3\}$
$\Xi_4 = \{x_4\}$
For $P_2 = (1,0,0,0)$ we have $\tau_2 = x$ and $q_2 = -(x_1 - 2)$. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2\}$
$\Xi_3 = \{x_3\}$

$\Xi_4 = \{x_4\}$
For $P_3 = (2, 1, 0, 0)$ we get $\tau_3 = x_2$ and $q_3 = x_2$. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2, x_2 - 1\}$
$\Xi_3 = \{x_3\}$
$\Xi_4 = \{x_4\}$
For $P_4 = (1, 1, 0, 0)$ we have $\tau_4 = x_1x_2$ and $q_4 = -(x_1 - 2)x_2$. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2, x_2 - 1\}$
$\Xi_3 = \{x_3\}$
$\Xi_4 = \{x_4\}$
For $P_5 = (2, 0, 1, 0)$ we get $\tau_5 = x_3$ and $q_5 = x_3$. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2, x_2 - 1\}$
$\Xi_3 = \{x_3, x_3 - 1\}$
$\Xi_4 = \{x_4\}$
For $P_6 = (1, 0, 1, 0)$ we have $\tau_6 = x_1x_3$ and $q_6 = -x_3(x_1 - 2)$. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2, x_2 - 1\}$
$\Xi_3 = \{x_3, x_3 - 1\}$
$\Xi_4 = \{x_4\}$
For $P_7 = (2, 0, 0, 2)$ we get $\tau_7 = x_4$ and we do not need to compute the triangular polynomial since τ_7 contains the maximal variable. The lists of factors are
$\Xi_1 = \{x_1 - 2, x_1 - 1\}$
$\Xi_2 = \{x_2, x_2 - 1\}$
$\Xi_3 = \{x_3, x_3 - 1\}$
$\Xi_4 = \{x_4, x_4 - 2\}$
For $P_8 = (1, 1, 0, 2)$ we get $\tau_8 = x_1x_4$ and we do not need to compute the triangular polynomial since τ_8 contains the maximal variable.
This time we have to stop. Indeed, we have to compare the x_2-ranges of P_4 and P_8 and, as one can see by the tower structure drawn above, the test fails.
We can then keep the computed factors and use the Association procedure to produce:

$$(x_1 - 1)(x_1 - 2), x_2(x_2 - 1), x_2x_3, x_3(x_3 - 1)$$

but, in order to finish, we have to switch to the original Axis of Evil algorithm:

$$x_4(x_2 + x_1 - 2), x_3x_4, x_4(x_4 - 2).$$

Part III

The Bar-Code language and some applications.

CHAPTER 5

The Bar-Code.

5.1 Introduction.

In this chapter, we define the main tool of this thesis: the *Bar-Code diagram* associated to a set of terms M.
In chapter 1, we defined two graphical representations for an M:

1. the diagrams introduced by M.G. Marinari and L. Ramella for terms in $3, 4, 5$ variables, which are particularly useful when dealing with problems involving terms arranged by degree (1.5);

2. the pictures with towers and the towers structures, which have been used connecting points and terms (1.4).

Actually, these representations can be handled only if we have a small number of terms and variables, otherwise the pictures become too complicated, if not impossible: *how to draw, for example, a* 5*-dimensional picture with towers?*
For our studies, we usually have to represent the Groebner escalier of a monomial ideal J.

If J is not zerodimensional, however, $\mathsf{N}(J)$ is an infinite set, so it becomes very difficult to draw it: *how to bridle the infinite?*
In order to break through this impasse, we introduced the *Bar-Code diagram*, which is a *bidimensional* picture mirroring exactly all the properties of the potential n-dimensional picture described above for any given set of terms M. The Bar-Code flattens everything in dimension 2 (simpler to handle) and is also very easy to draw.
Starting with the finite case, we will see how to connect such a picture to M and how to read properties directly by it.
After that, we will define *infinite Bar Codes*, in order to represent infinite set of terms.
Then, we will start dealing with applications of this construction, which turn out to be mainly combinatorial.

5.2 What is a Bar-Code? The finite case.

In this section we explain how to construct a Bar-Code diagram.
First of all, we associate to each term $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathcal{T}$ a list of n terms (one for each variable in $\mathcal{P}$). More precisely, for each $i \in \{1, ..., n\}$, we let

$$P_{x_i}(\tau) := x_i^{\alpha_i} \cdots x_n^{\alpha_n} \in \mathcal{T}, \text{ i.e. } P_{x_i}(\tau) = \frac{\tau}{x_1^{\alpha_1} \cdots x_{i-1}^{\alpha_{i-1}}}.$$

We can extend this procedure to a finite set of terms $M \subset \mathcal{T}$, defining, for each $i \in \{1, ..., n\}$,

$$M^{[i]} := P_{x_i}(M) := \{\sigma \in \mathcal{T} \mid \exists \tau \in M, P_{x_i}(\tau) = \sigma\}.$$

These operations on a term τ will play a fundamental role for the construction of the Bar-Code diagram.
Here we list some useful features.

1. For each $\tau \in \mathcal{T}$, $P_{x_1}(\tau) = \tau$.
2. If $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, $\alpha_i = deg_i(\tau) = 0$ then $P_{x_i}(\tau) = x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n}$.
3. The lex inequalities are maintained:
$$\tau <_{Lex} \sigma \Rightarrow P_{x_i}(\tau) \leq P_{x_i}(\sigma).$$
4. For each term τ and for any couple of indices i, j, say $1 \leq i < j \leq n$ we have
$$x_i \leq x_j \Rightarrow P_{x_j}(P_{x_i}(\tau)) = P_{x_i}(P_{x_j}(\tau)) = P_{x_j}(\tau).$$

Example 5.2.1. In $\mathbf{k}[x_1, x_2, x_3]$ consider $\tau = x_1x_2^3x_3^4$.
Clearly $P_{x_1}(\tau) = x_1x_2^3x_3^4$, while $P_{x_2}(\tau) = x_2^3x_3^4$ and $P_{x_3}(\tau) = x_3^4$.
For $\sigma_1 = x_2x_3^5 >_{Lex} \tau$, $P_{x_2}(\tau) = x_2^3x_3^4 <_{Lex} P_{x_2}(\sigma_1) = x_2x_3^5$.
For $\sigma_2 = x_1^5x_2^3x_3^4 >_{Lex} \tau$, $P_{x_2}(\tau) = x_2^3x_3^4 = P_{x_2}(\sigma_2)$.
$P_{x_3}(P_{x_2}(\tau)) = P_{x_2}(x_2^3x_3^4) = x_3^4 = P_{x_2}(P_{x_3}(\tau))$.

Now $M \subseteq \mathcal{T}$ will be a finite list of terms increasingly ordered w.r.t. lex.

Proposition 5.2.2. With the previous notation, if M is an order ideal in $\mathcal{T}$ then, for each $1 < i \leq n$, $M^{[i]}$ is an order ideal in $\mathcal{T}[i, n]$.

Proof: It is sufficient to prove the statement for $i = 2$; the general case can be brought back to this one by changing the indices of the variables.
For each $\sigma \in M^{[2]}$ and $\upsilon \mid \sigma$ we have $\upsilon \in M^{[2]}$. Namely, by definition of $M^{[2]}$ there exists $\tau \in M$ such that $\tau = x_1^{\alpha_1}\sigma$. Clearly $\upsilon \mid \tau$, so that $\upsilon \in M$ and $\upsilon = P_{x_2}(\upsilon) \in M^{[2]}$. $\diamond$

The following examples show that the converse of proposition 5.2.2 does not hold.

Example 5.2.3. In $\mathbf{k}[x_1, x_2, x_3]$, the set $M = M^{[1]} = \{1, x_1, x_2, x_1x_2, x_1^2x_2\} \subset \mathbf{k}[x_1, x_2, x_3]$ is not an order ideal, since $x_1^2x_2 \in M$ but $x_1^2 \notin M$, $x_1^2 \mid x_1^2x_2$.
Yet $M^{[3]} = \{1, 1, 1, 1, 1\}$, and $M^{[2]} = \{1, 1, x_2, x_2, x_2\}$ (seen as sets, so removing repeated elements) are order ideals.

Example 5.2.4. The set $M = M^{[1]} = \{1, x_1, x_2, x_3, x_1x_3, x_2x_3, x_2^2x_3\}$ is not an order ideal, since $x_2^2x_3 \in M$ and $x_2^2 \notin M$, as well as $M^{[2]} = \{1, 1, x_2, x_3, x_3, x_2x_3, x_2^2x_3\}$ ($x_2^2x_3 \in M^{[2]}$, while $x_2^2 \notin M^{[2]}$), whereas $M^{[3]} = \{1, 1, x_3, x_3, x_3, x_3\}$ is an order ideal.

Basing on the properties stated above, we construct a picture associated to a list $M = [\tau_1, ..., \tau_m]$.

Description 5.2.5. The *Bar Code* (or, simply, B-C) $\mathsf{B} := \mathsf{B}_\mathsf{M}$ of M is a "matrix", obtained in the following way.
We construct a $(n + 1) \times m$ table, containing ordinately the terms of M in the 0-th row, and in the (i, j) position, for $1 \leq i \leq n$, $1 \leq j \leq m$, the term $P_{x_i}(\tau_j)$.
The first row contains then the terms in $M^{[1]}$, i.e. the given elements of M, the second row

contains the terms in $M^{[2]}$ and so on:

$$\begin{array}{ccc} \tau_1 & ... & \tau_n \\ P_{x_1}(\tau_1) & ... & P_{x_1}(\tau_M) \\ P_{x_2}(\tau_1) & ... & P_{x_2}(\tau_M) \\ \vdots & & \vdots \\ P_{x_n}(\tau_1) & ... & P_{x_n}(\tau_M) \end{array} .$$

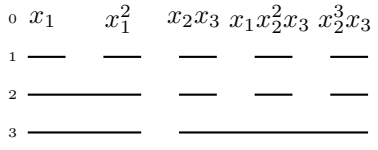
If a row contains some repeated terms, they are adjacent.
We replace the above "matrix" with an $(n+1) \times m$ table, constructed as follows.
The first row contains the ordered terms of M (assumed not to contain repeated terms). The second row (corresponding to $M^{[1]}$) contains as many segments as the elements of M .
The i-th row (corresponding to $M^{[i-1]}$) contains as many segments as the distinct elements of $M^{[i-1]}$ in such a way that to a set of r equal elements in $M^{[i-1]}$ corresponds a *unique* segment of *length* r, for each $2 \leq i \leq n$.

The segments composing the i-th line are called x_i*-bars* or, simply, i*-bars*.

Remark 5.2.6. Point out that we required not to have repeated elements in M in order for 1-bars all to have the same length, that we set as *unitary*.
Therefore, from now on, we suppose to have always finite lists of *distinct terms*.

Example 5.2.7. Given $M = \{x_1, x_1^2, x_2x_3, x_1x_2^2x_3, x_2^3x_3\} \subset \mathbf{k}[x_1, x_2, x_3]$, we have:
$M^{[1]} = \{x_1, x_1^2, x_2x_3, x_1x_2^2x_3, x_2^3x_3\}$
$M^{[2]} = \{1, 1, x_2x_3, x_2^2x_3, x_2^3x_3\}$
$M^{[3]} = \{1, 1, x_3, x_3, x_3\}$,
leading to the 4×5 table on the left and then to the B-C on the right:

$$\begin{array}{ccccc} x_1 & x_1^2 & x_2x_3 & x_1x_2^2x_3 & x_2^3x_3 \\ x_1 & x_1^2 & x_2x_3 & x_1x_2^2x_3 & x_2^3x_3 \\ 1 & 1 & x_2x_3 & x_2^2x_3 & x_2^3x_3 \\ 1 & 1 & x_3 & x_3 & x_3 \end{array}$$



We now give a formal definition for the concept of *range* introduced in 1.4.

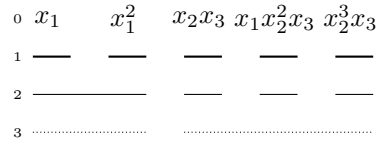
Definition 5.2.8. Given $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$, for any $\tau_j \in M$ ($j = 1, ..., m$), and $1 \leq i \leq n$, the x_i*-range* of τ_j in M is the set

$$R(i, \tau_j) := \{\sigma \in M | P_{x_i}(\sigma) = P_{x_i}(\tau_j)\}.$$

We will consider as representative for a range $R(i, \tau_j)$ its minimal element w.r.t. lex.
By construction, there is a one to one correspondence between ranges and bars in each line.

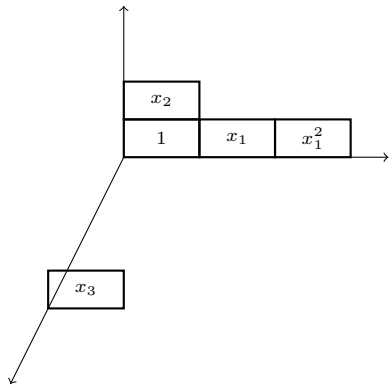
Example 5.2.9. Given $M = \{x_1, x_1^2, x_2x_3, x_1x_2^2x_3, x_2^3x_3\}$ as in example 5.2.7, the bars in B_M (read from the left to the right) correspond to the ranges in the following way:

- first line (thick in the picture below): $R(1, x_1), R(1, x_1^2), R(1, x_2x_3)$, $R(1, x_1x_2^2x_3), R(1, x_2^3x_3)$;
- second line (thin in the picture below): $R(2, x_1)$, $R(2, x_2x_3)$, $R(2, x_1x_2^2x_3)$, $R(2, x_2^3x_3)$;
- third line (dotted in the picture below): $R(3, x_1)$, $R(3, x_2x_3)$.



Up to now, we represented each term $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathcal{T}$, as a point in the n-dimensional space, considering the k-th exponent α_k as the k-th coordinate, $k = 1, ..., n$ of the corresponding point P_τ (see section 1.4). We point out that this representation mirrors the range's subdivision.

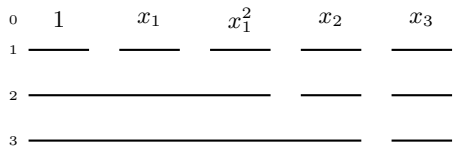
Example 5.2.10. Given $\mathsf{N} = \{1, x_1, x_1^2, x_2, x_3\} \subseteq \mathbf{k}[x_1, x_2, x_3]$, we get the picture below:



As in section 1.4, we notice that the single rectangles correspond to the x_1-ranges.
We select as many "planes" as the x_3-exponents of the elements in N. On the "plane" corresponding to x_3^0 we group the elements in N in horizontal lines according to their x_2-exponent so that on the bottom line lie the terms of N which are pure powers of x_1, in increasing order w.r.t. lex and in the higher line the terms having 1 as x_2-exponent, and so on.
A similar procedure is followed in the remaining x_3-planes.

Such a representation can become very complicated, when the number of the points and/or the variables increases, i.e. when dealing with a large number of points, and/or high-dimensional spaces. If we want to keep track of the range subdivision and of the properties of the terms which can be read by their mutual position in the n-dimensional space, we pass to the corresponding B-C. This is always a *bidimensional picture* but all the information on the terms is stored there.

Example 5.2.11. Considering again the set $M = \{1, x_1, x_1^2, x_2, x_3\} \subseteq \mathbf{k}[x_1, x_2, x_3]$ of example 5.2.10 we can easily draw $\mathsf{B_M}$, the corresponding B-C:



The 1-bars represent the single terms. The 2-bars group together the terms which were represented before as *grouped horizontally rectangles*. Finally, the 3-bars include all the terms whose corresponding rectangles lie in the same plane.
Point out that M is an order ideal.

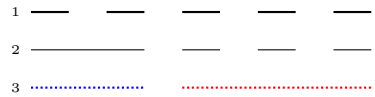
Now, we describe the properties of Bar-Codes, in order to characterize the corresponding pictures.

Definition 5.2.12. An *n-B-C* diagram B consists of n superimposed, horizontal lines, fragmented in segments called *bars*. Lines and bars are numbered from the top to the bottom and from the left to the right. Bars are such that

1. the bars composing the i-th row are called i-bars;
2. for each $1 \leq i \leq n-1$, under each i-bar in B lies at most one $(i+1)$-bar of B;
3. the 1-length of each 1-bar in B is conventionally set equal to 1;
4. for each $1 \leq j < i \leq n$ and for each i-bar A in B, the *length* of A w.r.t. j (shortly, the j-length of A) is the number of j-bars in B lying above A and is denoted by $l_j(A)$; the 1-length of A is simply called *length* of A and is denoted by $l(A)$;
5. for each $1 \leq i \leq n$ the sum of the lengths of the i-bars is the same.

Therefore, if a bar C lies under a bar D, $l(C) \geq l(D)$.
For each $1 \leq i \leq n$, we denote by $A_1^{(i)}, ..., A_{\mu(i)}^{(i)}$ the i-bars.
We call *bar list* of a Bar-Code B the list $\mathsf{L_B} := (\mu(1), ..., \mu(n))$, i.e. the list reporting the number of segments composing each row in B.

Example 5.2.13. Let B be the B-C.



Consider for example the dotted line. It is composed of two bars A_1, A_2 (A_1 is the blue bar, whereas A_2 is the red bar). We have $l_1(A_1) = 2$, $l_1(A_2) = 3$ and $l_2(A_1) = 1$, $l_2(A_2) = 3$.

Remark 5.2.14. 1) Conditions 1-5 of definition 5.2.12, mirror the properties of the P_{x_i}.

2) Given B, an n-B-C, one gets an $(n-h)$-B-C erasing h lines of B.

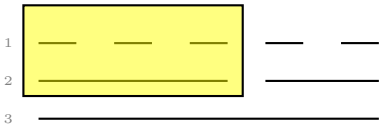
3) Fixed any i-bar ($2 \leq i \leq n$) of an n- B-C, the bars of the first $i-1$ lines lying above it form an $(i-1)$-(sub) B-C.

4) $\mu(1) \geq \mu(2) \geq ... \geq \mu(n)$.

Given a Bar Code B:

- a sub-Bar Code of B is the set B′ obtained by extracting some (even non-consecutive) lines from B;
- for every $1 \leq l < n$, an *l-block* associated to a bar A of B is the set containing A itself and all the bars of the $(l-1)$ lines lying immediately above A.

Example 5.2.15. In the Bar-Code B displayed below, the outlined part is a 2-block, namely the one associated to $A_1^{(2)}$.



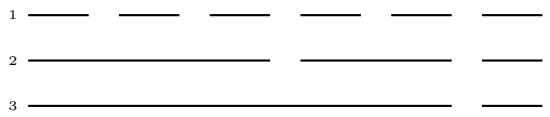
We come now to a turning point of our deal, since we need to associate to a given n-Bar-Code B, a finite set of terms M_{B}, with $\mathsf{B} = \mathsf{B}_{M_{\mathsf{B}}}$.
In order to achieve this goal follow the rules below:

Bc1. Let $\{A_1^{(n)}, ..., A_{\mu(n)}^{(n)}\}$, denote the n-bars of B and let $l_1(A_1^{(n)}) = l_0$, ..., $l_1(A_{\mu(n)}^{(n)}) = l_{\mu(n)}$. Substitute $A_1^{(n)}$ with l_0 copies of a random pure power $x_n^{a_0}$, $A_2^{(n)}$ with l_1 copies of a pure power $x_n^{a_1}$, $a_1 > a_0, \ldots$, $A_{\mu(n)}^{(n)}$ with $l_{\mu(n)-1}$ copies of a pure power $x_n^{a_{\mu(n)}}$, $a_{\mu(n)-1} > a_{\mu(n)-2}$.

Bc2. Take lines $i, i+1$, $i = 1, ..., n-1$ and construct all the possible blocks.
Repeat the construction inductively on the blocks, multiplying each term obtained in the i-th line ($1 \leq i < n$) by the term corresponding to the bar lying under it.

These two rules produce exactly the P_{x_i}'s for some M_{B}, so that operating on M_{B} according to description 5.2.5, we obtain back B.
Notice that the sets of terms which can be produced using Bc1 and Bc2 on a Bar-Code B are *infinite*. Indeed, we can start in Bc1 with *any* power of x_n and we can increase such a power by any natural number, while passing to a subsequent n-bar and the same reasoning can be applied to each inductive step, as shown in the example below.

Example 5.2.16. Consider the following B:



Two consistent sets of terms associated to B are, for example:

$$\begin{array}{cccccc} x_2 & x_1^3x_2 & x_1^5x_2 & x_2^5 & x_1x_2^5 & x_3^6 \\ x_2 & x_1^3x_2 & x_1^5x_2 & x_2^5 & x_1x_2^5 & x_3^6 \\ x_2 & x_2 & x_2 & x_2^5 & x_2^5 & x_3^6 \\ 1 & 1 & 1 & 1 & 1 & x_3^6 \end{array} \quad , \quad \begin{array}{cccccc} 1 & x_1 & x_1^2 & x_2 & x_1x_2 & x_3 \\ 1 & x_1 & x_1^2 & x_2 & x_1x_2 & x_3 \\ 1 & 1 & 1 & x_2 & x_2 & x_3 \\ 1 & 1 & 1 & 1 & 1 & x_3 \end{array}$$

that is $M_B = \{x_2, x_1^3x_2, x_1^5x_2, x_2^5, x_1x_2^5, x_3^6\}$ and $M'_B = \{1, x_1, x_1^2, x_2, x_1x_2, x_3\}$; note that M'_{B} is an order ideal whereas M_{B} is not.
In both cases, if we repeat on the 4×6 tables above the construction described in 5.2.5 we obviously get back to B.

Making Bc1 stricter, one gets BBc1, which can improve the properties of the resulting set $M_{\mathsf{B}} \subset \mathcal{T}$:

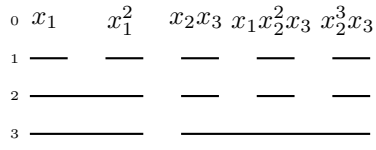
BBc1. Let $\{A_1^{(n)}, ..., A_{\mu(n)+1}^{(n)}\}$ be the n-bars of the given B, with $l_1(A_1^{(n)}) = l_0, ..., l_1(A_{\mu(n)}^{(n)}) = l_{\mu(n)}$. Substitute $A_1^{(n)}$ with l_0 copies of x_n^0, $A_2^{(n)}$ with l_1 copies of $x_n^1, \ldots$, $A_{\mu(n)}^{(n)}$ with $l_{\mu(n)-1}$ copies of $x_n^{\mu(n)-1}$.

Point out that BBc1 is simply a particular case of Bc1.

Example 5.2.17. Referring to example 5.2.16, the first set of terms associated to B can be obtained only if we apply Bc1, whereas the second is obtained using BBc1. This is the reason making order ideal the (unique!) set of terms obtained using BBc1 (see next lemma 5.2.23).

In this context, we need to point out that *we cannot associate an order ideal to every Bar-Code*.

Example 5.2.18. Given $M = \{x_1, x_1^2, x_2x_3, x_1x_2^2x_3, x_2^3x_3\}$ (which is not an order ideal) the associated B-C is $\mathsf{B_M}$:



which cannot be associated to *any* order ideal.
Using either Bc1, Bc2 or BBc1,Bc2, we obtain terms of the form:

$$\begin{array}{ccccc} x_1^{\alpha_1}x_2^{\beta_1}x_3^{\gamma_1} & x_1^{\alpha_2}x_2^{\beta_1}x_3^{\gamma_1} & x_2^{\delta_1}x_3^{\gamma_2} & x_2^{\delta_2}x_3^{\gamma_2} & x_2^{\delta_3}x_3^{\gamma_2} \\ x_2^{\beta_1}x_3^{\gamma_1} & x_2^{\beta_1}x_3^{\gamma_1} & x_2^{\delta_1}x_3^{\gamma_2} & x_2^{\delta_2}x_3^{\gamma_2} & x_2^{\delta_3}x_3^{\gamma_2} \\ x_3^{\gamma_1} & x_3^{\gamma_1} & x_3^{\gamma_2} & x_3^{\gamma_2} & x_3^{\gamma_2} \end{array},$$

with $\gamma_1 < \gamma_2$, $\delta_1 < \delta_2 < \delta_3$, $\alpha_1 < \alpha_2$ and so:

$$M_B = \{x_1^{\alpha_1}x_2^{\beta_1}x_3^{\gamma_1}, x_1^{\alpha_2}x_2^{\beta_1}x_3^{\gamma_1}, x_2^{\delta_1}x_3^{\gamma_2}, x_2^{\delta_2}x_3^{\gamma_2}, x_2^{\delta_3}x_3^{\gamma_2}\}.$$

If M_B were an order ideal, all the divisors of its elements should have to belong to M_B, so, even supposing $\gamma_1 = 0, \gamma_2 = 1$ and $\delta_1 = 0$, $\delta_2 = 1$, $\delta_3 = 2$, we would need to simultaneously have $\beta_1 = 0$, $\beta_1 = 1$, $\beta_1 = 2$, that is clearly impossible.
Actually the problem is that to a power of x_3 which is not the smallest one we associate three increasing powers of x_2 , whereas to the smallest power of x_3 we only associate 1 as power of x_2. This implies that any set of terms associated to the given B *cannot be an order ideal*, since some divisors are surely missing.

Inspired by example 5.2.18, we define *admissible Bar-Codes* as follows:

Definition 5.2.19. A Bar Code B is *admissible* if it exists at least one order ideal M_B.

A non-admissible B-C cannot be associated to an order ideal by definition, whereas the reverse does not hold, as we showed in example 5.2.16 where an admissible B-C is associated to a set $\mathsf{M_B}$ not satisfying the order ideal property.
A question then arises: *which are the admissible Bar Codes?*

Let B be a Bar Code and let M_B be the associated set of terms, via rules Bbc1 and Bc2. For each $i \in \{1, ..., n-1\}$, we fix a 3-block, composed of a $(i+2)$-bar A, all the $(i+1)$-bars $B_1, ..., B_h$ over A and all the i-bars over A[1].

[1]Such a condition is degenerate for $i = n-1$, since A would be an $(n+1)$-bar, so, for convenience, we imagine in the proof the whole diagram underlined by a unique "$(n+1)$-bar".

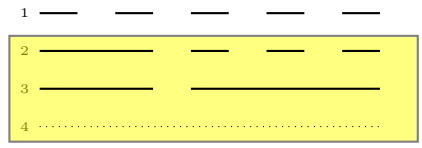
We check whether $l_i(B_j) \geq l_i(B_{j+1})$, $j = 1, ..., h-1$. If not, B is not admissible. If so, for $i = 3, ..., n$, fixed an $(i+1)$-bar we consider two consecutive i-blocks B_1, and B_2, lying over it and consisting of two consecutive i-bars A_1, A_2 and of all the bars lying above them. By the previous relation, $l_{i-1}(A_1) \geq l_{i-1}(A_2)$.

For each $j = 1, ..., l_{i-1}(A_2)$, we check $l_{i-2}(C_{1,j}) \geq l_{i-2}(C_{2,j})$, where $C_{1,j}, C_{2,j}$ are the j-th $(i-1)$-bars over A_1, A_2. If this test fails for some j, then the Bar-Code is not admissible. Then, isolated the $(i-1)$-blocks associated to $C_{1,j}, C_{2,j}$, we check the analogous property for all the couples of $(i-2)$-bars above the isolated blocks and so on. We prove now that if all the tests pass, then M_{B} is an order ideal.

If $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in M_{\mathsf{B}}$ and $x_i \mid \tau$ let $\tau' = \frac{\tau}{x_i} = x_1^{\alpha_1} \cdots x_i^{\alpha_i - 1} \cdots x_n^{\alpha_n}$.
We want to prove that in the original B there is a bar corresponding to τ' so that $\tau' \in M_{\mathsf{B}}$.
For each $1 \leq j \leq n$ let $A_{h_j}^{(j)}$ be the j-bar underlying τ. Since τ, τ' have the same n, $(n-1), ..., (i+1)$ exponents, if really $\tau' \in M_{\mathsf{B}}$, then it must lie over $A_{h_j}^{(j)}$ for $j = i+1, ..., n$. Additionally, τ' lies over $A_{h_i-1}^{(i)}$. Since $l_{i-1}(A_{h_i-1}^{(i)}) \geq l_{i-1}(A_{h_i}^{(i)})$, we can find the $(i-1)$-bar $A_l^{(i-1)}$ over $A_{h_i-1}^{(i)}$ corresponding to the exponent α_{i-1} of τ'. By the second test, the inequality also held for $A_l^{(i-1)}$ and $A_{h_{i-1}}^{(i-1)}$, so we can find a bar corresponding to the exponent α_{i-2} of τ'. By induction, we can conclude that $\tau' \in M_{\mathsf{B}}$.

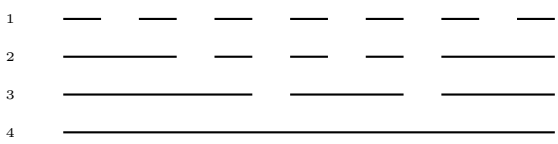
We now prove that conversely, if N is an order ideal, its associated Bar Code B_{N} passes the two tests above.
For each bar $C^{(i+2)}$, $i = 1, ..., n-1$[2], consider the associated 3-block denoting $B_1^{(i+1)}, ..., B_h^{(i+1)}$ the $(i+1)$-bars over $C^{(i+2)}$. If $\beta_1 = l_i(B_j^{(i+1)}) < l_i(B_{j+1}^{(i+1)}) = \beta_2$, for some $j < h$. By Bbc1, Bc2, $\sigma = x_i^{\beta_2 - 1} x_{i+1}^{j} x_{i+2}^{\alpha_{i+2}} \cdots x_n^{\alpha_n} \in \mathsf{N}$. But, since $\beta_1 = l_i(B_j^{(i+1)}) < l_i(B_{j+1}^{(i+1)}) = \beta_2$, the term $\sigma' = x_i^{\beta_2 - 1} x_{i+1}^{j-1} x_{i+2}^{\alpha_{i+2}} \cdots x_n^{\alpha_n} \notin \mathsf{N}$, and this contradicts the order ideal property, being $\sigma' \mid \sigma$.

Example 5.2.20. Example 5.2.18 shows a B-C failing the first test at the 3-blocks depending on the "degenerate" 4-bar:

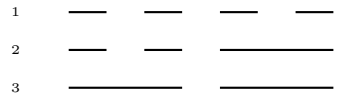


Example 5.2.21. Taken the following B, we show that the second test fails:

[2] Again we consider also the "degenerate" $(n+1)$-bar.



Even if we apply BBc1 and Bc2 to B we do not get an order ideal: the resulting set is indeed $\mathsf{M_B} = \{1, x_1, x_2, x_3, x_2x_3, x_3^2, x_1x_3^2\}$ and $x_1x_3^2 \in \mathsf{M_B}$, whereas $x_1x_3 \notin \mathsf{M_B}$.
Focus on the second and the third line and consider the blocks associated to $A_2^{(3)}$ and $A_3^{(3)}$, namely:



the fact that $x_1x_3^2 \in \mathsf{M_B}$, whereas $x_1x_3 \notin \mathsf{M_B}$ is mirrored by the fact that $l_1(A_3^{(2)}) < l_1(A_5^{(2)})$.

Remark 5.2.22. Consider a finite set of distinct terms $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$ and fix an $i \in \{1, ..., n\}$. For $j = 2, ..., m$, compute $\tau_{j,j-1} = \frac{\tau_j}{GCD(\tau_j, \tau_{j-1})}$.
If M is an order ideal, then $\tau_{j,j-1}$ consists of a unique variable by definition.
If $\max(\tau_{j,j-1}) > x_i$, then τ_{j-1}, τ_j do not lie over the same i-bar.

By rules BBc1 and Bc2 the following holds trivially.

Lemma 5.2.23. If B is an admissible B-C there is only one order ideal $\mathsf{M_B}$ associated to it.

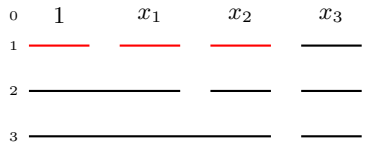
In the remaining sections, we will mostly deal with admissible Bar Codes, even if we will have some applications in which this property will not be required.

5.3 The star set.

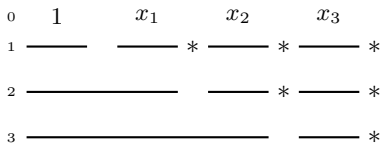
We are going to associate to a finite order ideal $\mathsf{N} \subseteq \mathcal{T}$ a new set of terms, arising from its admissible Bar Code $\mathsf{B_N}$. Rather loosely, these terms appear in correspondence with the ends of the rows from 1 to n and with some "holes" inside the rows from 1 to $n-1$.
In this section, $\mathsf{N} = \{\tau_1, ..., \tau_r\}$ will be an order ideal, $\mathsf{B} = \mathsf{B_N}$ the associated admissible Bar Code and I the monomial ideal such that $\mathsf{N}(I) = \mathsf{N}$.
First of all, we put a star at the end of each row of B as identification mark. We also put the same mark at each "hole" (between two consecutive bars) lying above a "hole" of the next line (hence no such star occurs in the last row).
Finally, we associate to each star a term, for instance to a star lying in the i-th row and after the j-th column we associate $x_iP_{x_i}(\tau_j)$ (thus to the end of the i-th row we associate

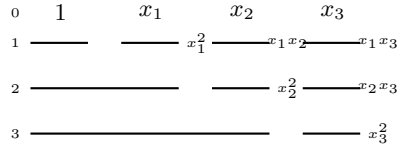
$x_i P_{x_i}(\tau_r)$).
We denote by $\mathcal{F}_{\mathsf{N}}$ the obtained set of terms and call it the *star set* of N.
We will call *Bar Code pictures* the Bar Codes equipped with the star set.

Example 5.3.1. Given the order ideal $\mathsf{N} = \{1, x_1, x_2, x_3\} \subset \mathbf{k}[x_1, x_2, x_3]$, so that $\mathsf{N} = \mathsf{N}(I)$ with $I = (x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2)$, its admissible B-C is B:



The "hole" between $A_1^{(1)}$ and $A_2^{(1)}$ does not lie above a hole of the second row, so we do not associate any star to it; on the other hand, in the hole between $A_2^{(1)}$ and $A_3^{(1)}$ we put a star to which we associate x_1^2.
Continuing this way along all B, we obtain the following two pictures





for which $\mathcal{F}_{\mathsf{N}} = \{x_1^2, x_1x_2, x_1x_3, x_2^2, x_2x_3, x_3^2\}$.
Note that in this case it holds $\mathcal{F}_{\mathsf{N}} = \mathsf{G}(I)$.

Next example shows that in general the star set $\mathcal{F}_N$ does not coincide with the minimal generating set of the monomial ideal I.

Example 5.3.2. Given the order ideal $\mathsf{N} = \{1, x_1, x_2, x_2^2, x_3\} \subset \mathbf{k}[x_1, x_2, x_3]$, so that $\mathsf{N} = \mathsf{N}(I)$ with $I = (x_1^2, x_1x_2, x_2^3, x_1x_3, x_2x_3, x_3^2)$, the corresponding admissible Bar Code, equipped with the star set is and $\mathcal{F}_{\mathsf{N}} = \{x_1^2, x_1x_2, x_1x_2^2, x_1x_3, x_2^3, x_2x_3, x_3^2\} \supsetneq \mathsf{G}(I)$.

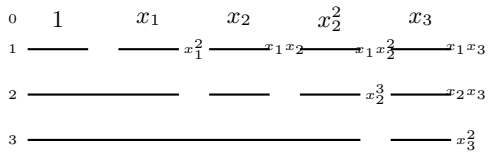


Figure 5.1: A Bar Code picture.

In Janet's context, for a monomial ideal I, it arises the set

$$\mathcal{F}(I) = \{x^\alpha \in \mathcal{T} \setminus \mathsf{N}(I) \,|\, \frac{x^\alpha}{\min(x^\alpha)} \in \mathsf{N}(I)\}$$

(especially connected with the so called *involutive bases*, see chapter 6).
As a matter of fact we can prove:

Proposition 5.3.3. With the above notation $\mathcal{F}_\mathsf{N} = \mathcal{F}(I)$.

Proof: First of all, we prove $\mathcal{F}_\mathsf{N} \subseteq \mathcal{F}(I)$.
Let σ be the term corresponding to the star between $P_{x_i}(\tau_j)$ and $P_{x_i}(\tau_{j+1})$, for $\tau_j, \tau_{j+1} \in \mathsf{N}$. Then, $\sigma = x_i P_{x_i}(\tau_j)$ by definition and $deg_i(\sigma) > \deg_i(\tau_j)$, but $deg_h(\sigma) = deg_h(\tau_j)$, for each $h > i$.

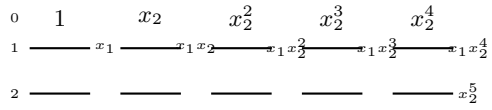
We now show that $\sigma \notin \mathsf{N}$.
If $\sigma \in \mathsf{N}$, then it must lie over the same $(i+1)$-bar as τ_j, but over the subsequent i-bar w.r.t. to the i-bar associated to τ_j, which cannot exist since σ arises from a star.
On the other hand, $\frac{\sigma}{\min(\sigma)} = P_{x_i}(\tau_j) \mid \tau_j$, so $P_{x_i}(\tau_j) \in \mathsf{N}$ by definition of order ideal. Thus $\sigma \in \mathcal{F}(I)$.

We prove now that $\mathcal{F}(I) \subseteq \mathcal{F}_\mathsf{N}$.
Let $\sigma \in \mathcal{F}(I)$, $\min(\sigma) = x_k$, so that $\frac{\sigma}{x_k} \in \mathsf{N}$. Let A be the k-bar of $\frac{\sigma}{x_k}$ and let $\tau_l \in \mathsf{N}$ be the rightmost element lying over A (so that $deg_h(\tau_l) = deg_h(\frac{\sigma}{x_k})$, $h = k, ..., n$).
We have to examine the terms τ_l, $\tau_{l+1} \in \mathsf{N}$.
First of all, notice that $\tau_{l+1} >_{Lex} \tau_l$ by assumption and it cannot be that $deg_h(\tau_{l+1}) = deg_h(\tau_l)$, $h = k, ..., n$ since, if it were so, τ_l would not have been the rightmost term lying over the k-bar A.
If $deg_k(\tau_{l+1}) > deg_k(\tau_l)$ and $deg_j(\tau_{l+1}) = deg_j(\tau_l)$, $j = k+1, ..., n$, then it would be that $\sigma \in \mathsf{N}$, contradicting $\sigma \in \mathcal{F}(I)$.
Thus, the $(k+1)$-bar underlying A breaks in correspondence to the end of A itself, so $\sigma \in \mathcal{F}_\mathsf{N}$.
$\diamond$

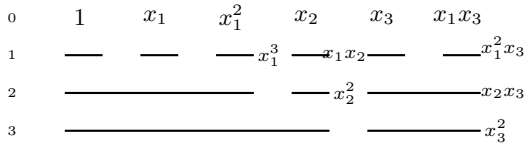
Thanks to proposition 5.3.3, by abuse of notation, we will call star set both $\mathcal{F}_\mathsf{N}$ and $\mathcal{F}(I)$.

Remark/Definition 5.3.4. It holds $\mathsf{G}(I) \subseteq \mathcal{F}_\mathsf{N} \subseteq \mathsf{B}(I)$.
Since $\mathcal{F}_\mathsf{N} = \mathcal{F}(I)$, we have $\mathcal{F}_\mathsf{N} \subseteq \mathsf{B}(I)$ because of the definition of $\mathcal{F}(I)$ and in general this inclusion may be strict. Analogously $\mathsf{G}(I) \subseteq \mathcal{F}_\mathsf{N}$.
If $\mathcal{F}_\mathsf{N} = \mathsf{G}(I)$, we say that B_N is a *full* Bar Code.

Example 5.3.5. a) Consider the order ideal $\mathsf{N} = \{1, x_2, x_2^2, x_2^3, x_2^4\} \subset \mathbf{k}[x_1, x_2]$, so that $I = (x_1, x_2^5)$.
The associated B-C picture is:



We have $\mathcal{F}_{\mathsf{N}} = \{x_1, x_1x_2, x_1x_2^2, x_1x_2^3, x_1x_2^4, x_2^5\} = \mathsf{B}(I)$.
b) Let $\mathsf{N} = \{1, x_1, x_1^2, x_2, x_3, x_1x_3\} \subset \mathbf{k}[x_1, x_2, x_3]$, so that $I = (x_1^3, x_1x_2, x_2^2, x_1^2x_3, x_2x_3, x_3^2)$.
The associated Bar Code picture is:

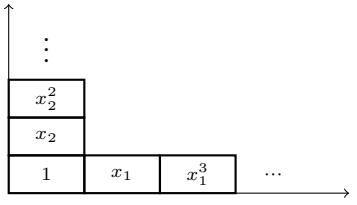


Since $\mathcal{F}_{\mathsf{N}} = \mathsf{G}(I) = \{x_1^3, x_1x_2, x_2^2, x_1^2x_3, x_2x_3, x_3^2\}$, B_{N} is a full B-C.
$\mathsf{B}(I) = \{x_1^3, x_1x_2, x_1^2x_2, x_2^2, x_1^2x3, x_2x_3, x_1x_2x_3, x_1^2x_3, x_3^2\} \supsetneq \mathcal{F}_{\mathsf{N}}$.

5.4 Infinite Bar Codes.

In this section we extend the notion of Bar Code to the case of non-zerodimensional monomial ideals.
We will first explain how to draw their B-C diagram, showing how to represent the *infinite* part of the Groebner escalier and then we will also study how to derive the star set from the B-C diagram.
If J is a non-zerodimensional monomial ideal we have $|\mathsf{N}(J)| = \aleph_0$ and still its minimal basis $\mathsf{G}(J)$ is a finite set, say $\mathsf{G}(J) = \{\sigma_1, ..., \sigma_r\}$.
We start examining the structure of $\mathsf{N}(J)$ in a very simple case.

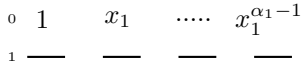
Example 5.4.1. Let $J = (x_1x_2) \triangleleft \mathbf{k}[x_1, x_2]$. In this simple case, we can represent the Groebner escalier $\mathsf{N}(J) = \{x_1^m,\ m \geq 0\} \cup \{x_2^l,\ l \geq 0\}$ in the plane:



If we examine the x_2-ranges composing $\mathsf{N}(J)$, we can observe that $R(2,1)$ is an infinite set, being $R(2,1) = \{x_1^m,\ m \geq 0\}$. Since $x_1x_2 \in \mathsf{G}(J)$, $R(2, x_2)$ is the singleton $R(2, x_2) = \{x_2\}$.

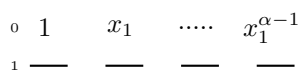
All the terms $x_1^{\alpha_1}x_2^{\alpha_2}$, $\alpha_1, \alpha_2 \geq 1$ belong to J, and no pure powers of x_2 belong to J, so, for each $i > 1$, $R(2, x_2^i)$ is the singleton containing x_2^i itself: therefore we have an infinite x_2-tower in the Groebner escalier.

We will draw the Bar Code inductively on the variables $x_1, ..., x_n$, using as a benchmark the monomial basis $\mathsf{G}(J)$.
In the case $n = 1$, if $J = (x_1^{\alpha_1})$ we have:



and, if $J = (0)$

where the symbol $\rightarrow$ stays for infinitely many 1 blocks, corresponding to the powers x_1^i, $i \in \mathbb{N}^*$ which belong to the (infinite) Groebner escalier.
Let us deal with the simple case $n = 2$.
a) Consider the set $\mathsf{G}(J) \cap \mathcal{T}[1]$, possibly containing the unique element of $\mathsf{G}(J)$, which is a pure power of x_1. We then distinguish $\mathsf{G}(J) \cap \mathcal{T}[1] \neq \emptyset$ and $\mathsf{G}(J) \cap \mathcal{T}[1] = \emptyset$.
In the first case, $x_1^\alpha \in \mathsf{G}(J)$, so $\mathsf{N}(J) \cap \mathcal{T}[1] = \{1, x_1, ..., x_1^{\alpha-1}\}$ and we draw its Bar-Code obtaining:



underlining it by a unique x_2-bar; we obtain a Bar-Code that we denote by B_1.
In the second case, no pure power of x_1 occurs in $\mathsf{G}(J)$, so we draw only the Bar-Code of term 1, putting after its single bar the symbol $\rightarrow$ underlining the obtained picture with a 2-bar: we denote again B_1 the obtained picture.
The symbol $\rightarrow$ stays for infinitely many 1 blocks, corresponding to the powers x_1^i, $i \in \mathbb{N}^*$ which belong to the (infinite) Groebner escalier.
b) Then we consider $\mathsf{G}(J) \setminus (\mathsf{G}(J) \cap \mathcal{T}[1])$, containing all the terms divisible by x_2, x_1x_2.
If it is the empty set, we put the symbol $\rightarrow$ after the 2-bar, the one drawn in B_1 before. Otherwise, we order its elements w.r.t. lex. Let τ_1 be the first element and let $deg_2(\tau_1) = e$.

We multiply the terms lying over B_1 by $x_2, ..., x_2^e$ and we copy, under them, the Bar-Code structures of B_1 e times.
On the first $(e-1)$ 2-bars we cannot have any multiple of a generator, since, in this case, there would be an element $\sigma \in \mathsf{G}(J) \setminus (\mathsf{G}(J) \cap \mathcal{T}[1])$ with $deg_2(\sigma) < e$.

The possible multiples of the generators will lie over the last 2-bar we drew. Considering this bar:

- if $\tau_1 = x_2^e$ we delete τ_1 and all the bars under it;
- if $\tau_1 = x_1^l x_2^e$ consider the 1-bars. More precisely, if x_2^e is followed by $\rightarrow$, we remove the symbol and we add the terms $x_1 x_2^e, ..., x_1^{l-1} x_2^e$, each one underlined by a 1-bar. Otherwise, we delete all the multiples of τ_1, checking the terms from right to left.

Then repeat this procedure for the other terms in the set, replacing B_1 with the last 2-block. At the end, we put a symbol $\rightarrow$ after the last 2-block if no pure powers of x_2 occur in the set. Let us see a first example

Example 5.4.2. Consider the monomial ideal $J = (x_1^2 x_2^2) \triangleleft \mathbf{k}[x_1, x_2]$.
We have $\mathsf{G}(J) \cap \mathcal{T}[1] = \emptyset$, so the first step produces the Bar Code B_1 below.

$$\begin{array}{c} 1 \\ \text{——}\rightarrow \\ \text{——} \end{array}$$

Then, we consider $\sigma \in \mathsf{G}(J) \setminus (\mathsf{G}(J) \cap \mathcal{T}[1]) = \{x_1^2 x_2^2\}$.
Since $deg_2(x_1^2 x_2^2) = 2$ we get:

$$\begin{array}{ccccc} 1 & & x_2 & & x_2^2 \\ \text{——}\rightarrow & & \text{——}\rightarrow & & \text{——}\rightarrow \\ \text{——} & & \text{——} & & \text{——} \end{array}$$

We consider the last 2-bar. Since we have the symbol $\rightarrow$, but $x_1^2 x_2^2 \in \mathsf{G}(J)$ we finally get:

$$\begin{array}{ccccc} 1 & & x_2 & x_2^2 & x_1 x_2^2 \\ \text{——}\rightarrow & & \text{——}\rightarrow & \text{——} & \text{——} \\ \text{——} & & \text{——} & \text{————} & \rightarrow \end{array}$$

We now state the general procedure for the case $n > 2$.
Suppose we have computed B_{h-1}, $2 \leq h \leq n$, involving the terms divisible only by $x_1, ..., x_{h-1}$. We find the first h-block by underlining B_{h-1} with the first h-bar.
Consider $\mathsf{G}[h] := (\mathsf{G}(J) \cap \mathcal{T}[h]) \setminus (\mathsf{G}(J) \cap \mathcal{T}[h-1])$.
If $\mathsf{G}[h]$ is empty, we put an $\rightarrow$ after the first h-bar (meaning that the first h-block repeats infinitely many times and at each repetition the terms over the previous copy are multiplied by x_h). Then, we underline the obtained picture with the first $(h+1)$-bar.
Otherwise, if $\mathsf{G}[h] \neq \emptyset$, we order it w.r.t. lex and, for each $\tau \in \mathsf{G}[h]$, by definition, $\max(\tau) =$

x_h. Denoting F the rightmost h-block we drew, we let $deg_h(\tau) = \alpha_h$ and β the maximal h-degree of the terms lying above F (of course $\beta = 0$ if τ is the smallest element of $\mathsf{G}[h]$). We write $\alpha_h - \beta$ copies of F and at each repetition the terms over the previous copy are again multiplied by x_h. If some multiple of τ appears among the terms inserted so far in the Bar Code picture it will lie above the $(\alpha_h - \beta)$-th copy of F, from now on denoted by $\widetilde{F}$ (since for the previous ones the h-degree is too small) and it has to be removed since it cannot belong to $\mathsf{N}(J)$, so we have to modify $\widetilde{F}$.

(a) If $\tau = x_h^{\alpha_h}$ we simply need to remove the whole $\widetilde{F}$.

(b) If $\tau = x_1^{\alpha_1} \cdots x_{h-1}^{\alpha_{h-1}} x_h^{\alpha_h}$, $(\alpha_1, ..., \alpha_{h-1}) \neq (0, ..., 0)$ let $x_i = \max(\frac{\tau}{x_h^{\alpha_h}}) < x_h$, we must distinguish two subcases:

(b1) $x_i = \min(\tau)$: for each i-bar of $\widetilde{F}$ we consider the related i-block. We must erase the possible multiples of τ and all the bars of $\widetilde{F}$ lying under them. In particular, if the i-block under consideration is followed by an $\rightarrow$, denoting γ the i-degree of the terms involved, we add $\alpha_i - \gamma - 1$ copies of our i-block erasing from them the possible multiples of τ and related bars, as well as the $\rightarrow$ (if $\alpha_i - \gamma - 1 < 0$ both the whole i-block and the $\rightarrow$ have to be deleted).

(b2) Otherwise we again consider the i-blocks of the i-bars of $\widetilde{F}$, distinguishing three possibilities for each i-block H.

1. The i-degree of the terms over H is smaller then α_i and H is not followed by an $\rightarrow$. In this case, no term over H is multiple of τ as its i-degree is too small, so H does not have to be modified.
2. The i-degree γ of the terms over H is smaller than α_i, but H is followed by an $\rightarrow$. In this case we remove the arrow and we make $\alpha_i - \gamma$ copies of H putting an $\rightarrow$ on the lower right-hand corner of the last copy $\widetilde{H}$. By construction, the terms lying above $\widetilde{H}$ have α_i as i-degree, so the possible multiples of τ should lie over it. We then compute $\max(\frac{\tau}{x_i^{\alpha_i} x_h^{\alpha_h}})$, repeating (b) for $\widetilde{H}$ and $\frac{\tau}{x_i^{\alpha_i} x_h^{\alpha_h}}$, until we reach $\min(\tau)$ (and we apply (b1) for it).
3. The i-degree of the terms over H is greater or equal than α_i. In this case computing $\max(\frac{\tau}{x_i^{\alpha_i} x_h^{\alpha_h}})$, we repeat the last part of 2.

We show that for each $\tau \in \mathsf{G}[h]$, steps (a),(b) ensure that the Bar Code picture we obtain does not contain τ. If τ is a pure power of x_h, by (a) clearly the Bar Code picture we obtain does not contain τ (as we have deleted the whole $\widetilde{F}$). If τ is not a pure power of x_h, step (b) essentially checks whether for each variable $x_i < x_h$, such that $x_i \mid \tau$, it can be that

an i-degree is greater or equal than α_i and deletes all the possible elements satisfying this condition.
At this point, the only possible multiples σ of τ that could appear in the Bar Code picture we drew treating τ are such that $\max(\sigma) = x_h$ and $deg_h(\sigma) = \alpha_h$ (up to this moment there does not appear in the picture neither terms containing the variables greater than x_h nor terms with maximal variable x_h and deg_h greater than α_h). If τ were a pure x_h-power we would have deleted all the $\widetilde{F}$ block (and so all the multiples of τ w.r.t. $x_1, ..., x_h$) by (a). If τ were not a pure x_h-power by (b) we would have deleted inside $\widetilde{F}$ all the terms having exponents greater or equal to those of τ w.r.t. $x_1, ..., x_h$. So no σ with $\tau \mid \sigma$, $\max(\sigma) = x_h$ and $deg_h(\sigma) = \alpha_h$ appears in the Bar Code picture obtained up to τ.
Possible multiples σ of τ with either $\max(\sigma) > x_h$ or $deg_h(\sigma) > \alpha_h$ should arise from terms of $\mathsf{G}(J)$ greater than τ. Let τ' be the term in $\mathsf{G}(J)$ next to τ and let H be the last H-block we got from τ, all of whose terms have x_h as maximal variable.
We distinguish three possibilities according to the part of the Bar Code we have to copy:

1. the last h-block H a finite number of times;

2. the whole Bar Code (i.e. we are constructing the first $(h+1)-$block);

3. a sub-Bar Code ending with an $\rightarrow$.

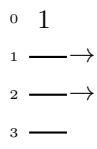
As for 1 we are adding h-blocks, incrementing the x_h-exponents of the involved terms keeping the other exponents fixed: this way we cannot get multiples of τ since the x_i-exponents $i = 1, ..., h-1$ are too small.
As for 2 we are introducing $(h+1)-$blocks incrementing the x_{h+1} exponents of the involved terms keeping fixed the exponents of $x_1, ..., x_h$ and again we cannot get multiples of τ.
As for 3 we are copying a sub-Bar Code ending with an $\rightarrow$. Since we have already seen how the exponents of x_h, x_{h+1} can increase (respectively in 1 and 2), we know that our arrow must refer to an i-block with $i = 1, ..., h-1$. Increasing the x_i-exponents again we must distinguish whether the i-block we are manipulating is a copy of something inside H (in which case some exponent of the variables smaller than x_h is too little) or not (this means that it lays on an h-block with x_h-exponent smaller than α_h).
Up to now we have seen that no terms in $\mathsf{T}(J)$ can appear in the Bar Code picture we are drawing. Finally, we show that each $\tau \in \mathsf{N}(J)$ actually appears in the Bar Code picture. We know that 1 actually appears, this implies that for each i-block the leftmost term over it does not contain variables smaller than x_i.
Let $\tau \in \mathsf{N}(J)$ and let $\max(\tau) = x_h$. By construction if τ belongs to the Bar Code picture it must lie over the first $h+1, ..., n$ bar. Let then $\alpha_h = deg_h(\tau)$ and let β be the maximal

h-degree of terms of the first $(h+1)-$bar (so that the last h-block we drew has an $\rightarrow$ or $x_h^{\beta+1} \in \mathsf{G}(J)$). Two possibilities can arise:

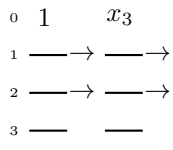
- $\alpha_h \leq \beta$: we move to the h-bar corresponding to α_h and look what happens for the x_{h-1}-exponent;
- $\alpha_h > \beta$ means that $x_h^{\beta+1} \notin \mathsf{G}(J)$ [54], thus after the last h-block there is an arrow. So there is $\sigma \in \mathsf{N}(J)$ with $deg_h(\sigma) = \beta$ and if σ is represented in $\mathsf{N}(J)$ also τ does. We look then for σ inductively on the variables.

Clearly if we have an arrow in the inductive step we do not have a pure power but also $x_h^{\alpha_h}$ and the intermediate variables which have already been fixed [54].
Let us see some significant examples.

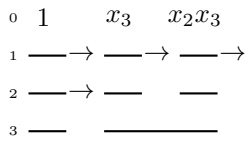
Example 5.4.3. Let $J = (x_2^2 x_3) \triangleleft \mathbf{k}[x_1, x_2, x_3]$. Assume we have computed the Bar Code B_2 of the terms divided only by x_1 and x_2, underlining B_2 by a first 3-bar. We have:



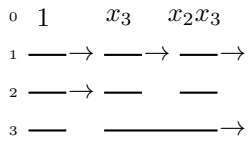
Consider $\mathsf{G}(J) \setminus (\mathsf{G}(J) \cap \mathcal{T}[2])$. Since it is not empty, we deal with its only element $\tau = x_2^2 x_3$. We have $deg_3(\tau) = 1$, so we make only one copy of the first 3-block and we get:



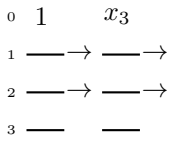
Now, $x_2 = \max(\frac{\tau}{x_3}) = \min(\tau)$, so we perform as in (b1), getting:



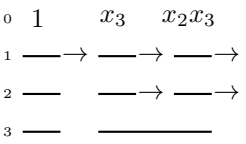
Since there are no pure powers of x_3, we finally get:



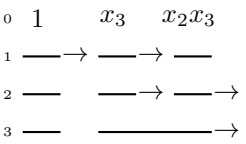
Example 5.4.4. Let $J = (x_1x_2x_3) \triangleleft \mathbf{k}[x_1, x_2, x_3]$. The first two steps are the same as in example 5.4.3 so we have:



Then, since $\max(\frac{\tau}{x_3}) = x_2 \neq \min(x_1x_2x_3)$ we first draw:

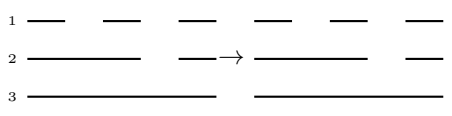


and finally, dealing with $x_1 = \min(\tau)$, we get the final picture:



Given a Bar Code B, we naturally extend to infinity the concept of j-length. The only difference is the presence of $\rightarrow$: if, over an $(i+1)$-bar C there are the i-bars $A_1, ..., A_k$ and after $A_k \rightarrow$ occurs, then $l_i(C) = \infty$.
Also for infinite Bar Codes we can define the analogous of rules Bbc1 and Bc2, in order to associate to them infinite sets of terms. The only difference is again represented by $\rightarrow$, in this case we write down as many terms as the 1 bars really drawn in the diagram, performing the same Bbc1 and Bc2 as in the finite case, so disregarding $\rightarrow$.
Again, the problem of admissibility arises, but it can be solved exactly as in the finite case, exploiting the extension to infinity of the length functions.

Example 5.4.5. Consider the infinite Bar Code B

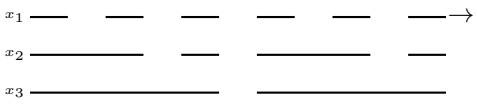


We have $l_2(A_1^{(3)}) = \infty$, $l_2(A_2^{(3)}) = 2$.
If we apply Bbc1, Bc2 we get

$$\begin{array}{cccccc} 1 & x_1 & x_2 & x_3 & x_1x_3 & x_2x_3 \\ 1 & 1 & x_2 \to & x_3 & x_3 & x_2x_3 \\ 1 & 1 & 1 & x_3 & x_3 & x_3 \end{array}$$

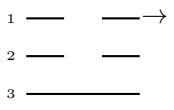
so $M_{\mathsf{B}} = \{1, x_1, x_3, x_1x_3, x_2x_3\} \cup \{x_2^m,\, m \geq 1\}$. Such a set is an order ideal and actually B passes both the admissibility tests generalized to infinity.

Example 5.4.6. The Bar Code



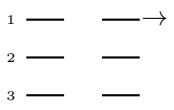
is not admissible, failing simultaneously both the tests.

Example 5.4.7. The Bar Code



is not admissible, since it fails the first test for the 3-block formed by the 3 block associated to the $A_1^{(3)}$.

Example 5.4.8. Consider the Bar Code displayed below.



It passes the first test, but it fails the second one: the comparison failing is the one between the 2-blocks over $A_1^{(3)}, A_2^{(3)}$.

As for the finite case, we can read the (infinite) star set $\mathcal{F}_{\mathsf{N}}$ directly from the Bar-Code. First of all, we consider the holes between two bars, not filled by $\to$ and we proceed as in the finite case (computing the $P_{x_{\_}(\_)}$'s of the last term before the hole) and we do the same also for the bars at the end of a line, if there is not the symbol $\to$.
The obtained terms belong to $\mathcal{F}_{\mathsf{N}}$ for the same reasons as for finite Bar Codes and we call them *finite terms*.

Proposition 5.4.9. With the notation above, the star set $\mathcal{F}_{\mathsf{N}}$ consists of the terms of the form

$$\tau x_{j_1}^{\alpha_{j_1}} \cdots x_{j_l}^{\alpha_{j_l}},\ \alpha_{j_i} \geq 0,$$

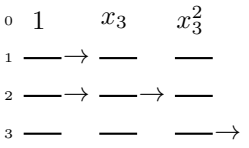
where:

- τ is a finite term in the h-th bar ($h \in \{1, ..., n\}$);

- x_{j_i}, $j_i > h$, s.t. on the j_j-bar, under the star corresponding to τ, there is the symbol "$\rightarrow$".

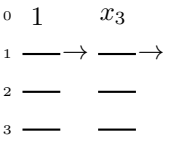
Proof: In what follows, we denote by J the monomial ideal such that $\mathsf{N} = \mathsf{N}(J)$.
If τ is a finite term, it belongs to $\mathcal{F}_{\mathsf{N}}$ by 5.3.3. If we consider a finite term $x_k\omega \in \mathcal{F}_{\mathsf{N}}$, $\min(x_k\omega) = x_k$ and we suppose to have the symbol $\rightarrow$ on a hole in the x_l-line ($l > k$) under the star corresponding to $x_k\omega \in \mathcal{F}_{\mathsf{N}}$, we get that $\omega x_l^m \in \mathsf{N}$ for each m and $x_k\omega x_l^n \in J$, being a multiple of $x_k\omega$. This implies $x_k\omega x_l^m \in \mathcal{F}_{\mathsf{N}}$ for each m.
This holds also in the case we have more than one variable displaying $\rightarrow$ under a finite term. The only difference is that we can increase the exponent of each of these variables.
Consider now $x_k\omega \in \mathcal{F}_{\mathsf{N}}$, $\min(x_k\omega) = x_k$. If it is a finite term we have nothing to prove. Suppose then that $x_k\omega$ is not a finite term. This means that $\omega \in \mathsf{N}$ is represented in the Bar Code in the repetition induced by $\rightarrow$ placed on one or more than one variable greater than x_k. Then, there is $\omega' \in \mathsf{N}$, i.e. the term followed by the symbols $\rightarrow$, obtained dividing it by these variables. The term $x_k\omega'$ is a finite term, so we can conclude. $\diamond$

Example 5.4.10. Let us consider the ideal $J = (x_1x_3, x_2x_3^2) \triangleleft \mathbf{k}[x_1, x_2, x_3]$.



We get $\mathcal{F}(J) = \{x_1x_2^mx_3,\ m \geq 0\} \cup \{x_1x_3^l,\ l \geq 2\} \cup \{x_2x_3^k,\ k \geq 2\}$.

Example 5.4.11. For the monomial ideal $J = (x_2, x_3^2) \triangleleft \mathbf{k}[x_1, x_2, x_3]$ we have



The star set is $\mathcal{F}_{\mathsf{N}} = \{x_2, x_2x_3, x_3^2\}$.
We point out that in those case the star set is *finite*, even if the Groebner escalier is *infinite*.

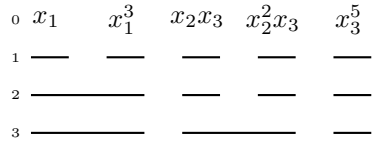
5.5 How to encode a Bar Code?

Given a finite set of terms $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$, we have to face the problem of *encoding* a Bar Code in a computer.
Indeed there are differences between the visual representation one can give to data and the way a computer stores them in memory.
The most suitable data structure which can be used to encode a Bar Code is the *trie structure*.
More precisely, we label the root with the set M, that we suppose, as usual, ordered w.r.t. lex. Each edge adjacent to the root represents an increasing P_{x_n} w.r.t. lex and we label each node at level 1 with the sets of terms sharing the same P_{x_n}'s. Continuing this way with $x_{n-1}, ..., x_1$ we get a trie in which the terms are arranged w.r.t. their P_{x_i}'s, grouping together at level $1 \leq i \leq n$ the ones whose P_{x_i}'s are the same.
Essentially, each edge represents a bar: the edges connecting level 0 to level 1 are the n-bars, the ones connecting level 1 to level 2 are the $(n-1)$-bars and so on.
This way, reading information from a Bar Code becomes the same as reading information from a trie.

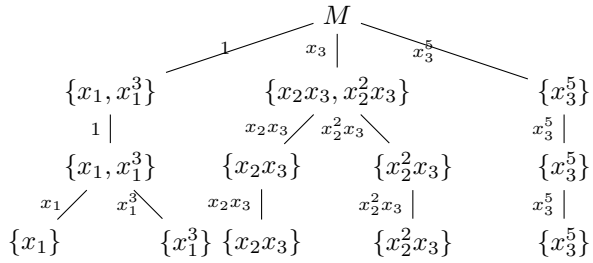
Example 5.5.1. For $M = \{x_1, x_1^3, x_2x_3, x_2^2x_3, x_3^5\} \subseteq \mathbf{k}[x_1, x_2, x_3]$, we have:

$$\begin{array}{ccccc} x_1 & x_1^3 & x_2x_3 & x_2^2x_3 & x_3^5 \\ x_1 & x_1^3 & x_2x_3 & x_2^2x_3 & x_3^5 \\ 1 & 1 & x_2x_3 & x_2^2x_3 & x_3^5 \\ 1 & 1 & x_3 & x_3 & x_3^5 \end{array}$$

so the Bar Code is:



and we encode it as



As seen in description 5.2.5, we have to compute the $P_{x\_}(\_)$ of each $\tau_j \in M$, $j = 1, ..., m$ and τ_j, τ_{j+1} lie over the same i-bar if $P_{x_i}(\tau_j) = P_{x_i}(\tau_{j+1})$, for $j = 1, ..., m-1$, $i = 1, ..., n$.
Since M is ordered w.r.t. lex, thus possible repeated $P\_(\_)$ are adjacent, we can perform the construction as follows:

- read the x_n exponents and arrange the terms into the x_n-ranges, allocating the first level of the trie;
- for each node in the trie, read the x_{n-1} exponents and allocate the second level;
- repeat until x_1 is reached.

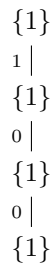
Such an encoding has complexity $O(nm)$, since, for each $\tau_j \in M$, $j = 1, ..., m$ we only have to read the exponents, one by one.
Now we discuss the next item, i.e. how to encode the Bar Code picture, adding the stars.
Since the bars are in correspondence with the edges in the trie, the construction we perform to settle the stars, costs computationally speaking, as attaching a new node to each node of level $0, ..., n-1$, so it is $O(nr)$, where $r+1$ is the maximal degree of a node in the trie.
The encoding of an infinite Bar Code is similar, but we label only the edges followed by $\rightarrow$ with an R, meaning that the corresponding bar (and the whole subtree depending on it) is repeated infinite times.
For the complexity of an infinite Bar Code, we notice that we have to deal at most with terms of degree d, where d is the sum of the maximal degrees of $x_1, ..., x_n$ in the terms of $\mathsf{G}(J)$ and we deal with them at most once for each variable.

5.6 A Bar-Code algorithm for a finite set of distinct points.

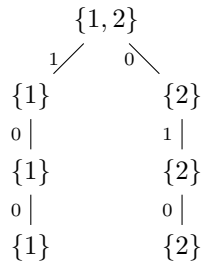
In this section, we describe how to compute the Groebner escalier N of the ideal $I(\mathbf{X})$ of a finite set of distinct points $\mathbf{X}$, setting a biunivocal correspondence between such points and the elements of N.
As explained in chapter 1, there are several algorithms dealing with this problem, such as, for example, Cerlienco-Mureddu Correspondence and the Lex Game.
The most important feature of Cerlienco-Mureddu Correspondence is its iterativity on the elements of $\mathbf{X}$, whereas the Lex Game (as the other methods described in chapter 1) is faster then Cerlienco-Mureddu algorithm but, requiring some preprocessing on $\mathbf{X}$, it is not iterative.
The algorithm developed here, places itself halfway between the Cerlienco-Mureddu Correspondence and the other methods. Indeed, our algorithm maintains Cerlienco-Mureddu's

iterativity but, thanks to the B-C structure, it shares some facilities with the Lex Game.
Let us consider a set $\mathbf{X} = \{P_1, ..., P_S\} \subseteq \mathbf{k}^n$, $P_i = (a_{i1}, ..., a_{in})$, for $i = 1, ..., S$ and define $\mathsf{N}(i) := \mathsf{N}(I(\{P_1, ..., P_i\})) = \{\tau_1, ..., \tau_i\}$, $\mathsf{B}(i) = \mathsf{B}_{\mathsf{N}(i)}$ and $\mathbf{X}_i = \{P_1, ..., P_i\}$.
First of all, we recall that $|\mathsf{N}(S)| = |N(\mathbf{X})| = |\mathbf{X}| = S$.
We can associate a trie $\mathfrak{T}(\mathbf{X})$ to the set $\mathbf{X}$. Such a trie is constructed iteratively on the points of $\mathbf{X}$ and $ht(\mathfrak{T}(\mathbf{X})) = n$ is the number of coordinates of each point.
The edges are labeled with the coordinates of the points. The root is labeled with the set $\{1, ..., S\}$, whereas a node at level l is labeled by the set $\{\alpha_1, ..., \alpha_h\}$, $\alpha_1 < ... < \alpha_h$, where $P_{\alpha_1}, ..., P_{\alpha_h} \in \mathbf{X}$ are the points whose first l coordinates are equal to the ones identified by the edges in the path from the root to the considered node.

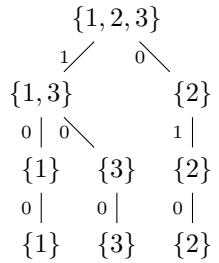
Example 5.6.1. Given the set $\mathbf{X} = \{(1,0,0), (0,1,0), (1,1,2), (1,0,3)\}$, we display here the construction of $\mathfrak{T}(\mathbf{X})$.
We start with $P_1 = (1,0,0)$, associating to it $\mathfrak{T}(\mathbf{X}_1)$:



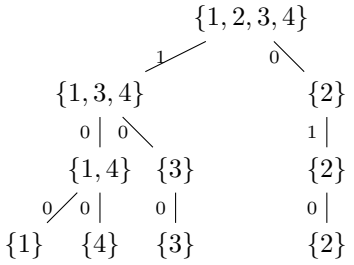
The second point $P_2 = (0,1,0)$ has no common coordinates with P_1, so $\mathfrak{T}(\mathbf{X}_2)$ is



The point $P_3 = (1,1,2)$ shares the first coordinate with P_1, so for $\mathfrak{T}(\mathbf{X}_3)$ we get

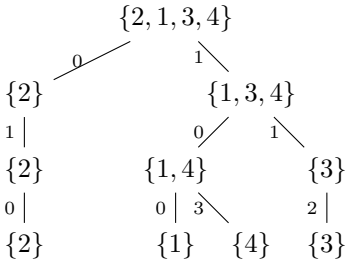


The point $P_4 = (1, 0, 3)$ shares the first two coordinates with P_1. The final trie $\mathfrak{T}(\mathbf{X}) = \mathfrak{T}(\mathbf{X}_4)$ is

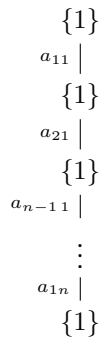


The sets labeling the nodes correspond to the classes Σ_i, $i = 0, ..., n$ in the Lex Game algorithm and in the Jumping algorithm but in this case they are not ordered w.r.t. any criterion. Indeed, their order depends only on the order of the elements in $\mathbf{X}$.

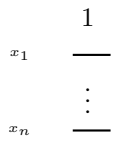
Example 5.6.2. Take the set $\mathbf{X}$ of example 5.6.1, but order the points in this way: $\mathbf{X} = \{P_2, P_1, P_3, P_4\}$.
For the set ordered this way, we get



The trie we constructed and the Bar Code are the main tools for our algorithm.
Let us explain the whole construction for $\mathbf{X} = \{P_1, ..., P_S\}$.
For $S = 1$ we construct $\mathfrak{T}(\mathbf{X}_1)$ and we set $\mathsf{N}(1) = \{1\}$.



The $\mathsf{B}(1)$ displayed below is the associated B-C:



The above construction for $i = 1$ has to be considered as the base step for the algorithm.
This step is correct since, if the given set is the singleton $\{P_1\}$, the associated ideal is the maximal ideal $I(\{P_1\}) = (x_1 - a_{1,1}, ..., x_n - a_{1,n})$ and so the Groebner escalier is clearly $\mathsf{N}(1) = \{1\}$.
We construct N inductively on $i = 2, ..., S$, associating a term τ_i to each P_i, through the following steps.

1. Set a list $D = [\underbrace{\emptyset, ..., \emptyset}_{m \text{ times.}}]$[3].
2. Construct $\mathfrak{T}(\mathbf{X}_i)$.
3. Compute the maximal integer s, such that $\Pi_{s-1}(P_i, \mathbf{X}) \neq \emptyset$, i.e. the level of the trie in which the path in $T(\mathbf{X}_i)$ corresponding to P_i forks[4].
4. Since τ_i will then belong to $R(s+1, 1)$, point to its corresponding bar, namely $A_1^{(s+1)}$.
5. Let L be the subset of the set of terms over $A_1^{(s+1)}$, consisting of all the terms τ_j corresponding to points P_j such that $\pi_{s-1}(P_i) = \pi_{s-1}(P_j)$. Then compute $\tau_l = Max_{Lex}(L)$ and keep track of the value l, calling it the $s-1$ *antecedent* of P_i.

[3] If we set $D[3] = 4$, we imagine $D = [\emptyset, \emptyset, 4, \underbrace{\emptyset, ..., \emptyset}_{m-3 \text{ times.}}]$

[4] In Cerlienco-Mureddu language, this integer is the σ-value of the point P_i; see [20, 21, 22] for more details.

6. Take the s-bar lying under τ_l, say $A_h^{(s)}$. The term τ_i has to lie over $A_{h+1}^{(s)}$ and also this bar lies over $A_1^{(s+1)}$. There are two different possibilities:

   a. this $A_{h+1}^{(s)}$ has not been constructed yet;

   b. this $A_{h+1}^{(s)}$ has already been constructed.

7. If a. occurs, set $\tau_i = x_s\tau_j$, where $\tau_j := Min(R(s, \tau_l))$ and update the B-C adding $A_{h+1}^{(s)}$ to it.

8. If b. occurs, move to $A_{h+1}^{(s)}$. Then

   (a) let $\mathbf{Y} = \{P_{\alpha_1}, ..., P_{\alpha_h}\}$ be the set of points corresponding to the terms lying over $A_{h+1}^{(s)}$;

   (b) set $D[\alpha_j] = 1$, $j = 1, ..., h$;

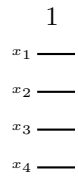
   (c) read the path of P_i in the coordinate trie, from level s to level 0, looking for the first node f whose label contains at least an element α_j (index of a point in $\mathbf{Y}$) in addition to i: $f + 1$ is the new σ-value.
   Browsing the elements of the node's label keeping the left, the f-antecedent of P_i is its nearest element not sharing more than f coordinates with any of the remaining points of the label (so we are in the first $s - 1, ..., f + 2$ bar).

   (d) Repeat the steps $6 - 8$, until level 0 has been reached.

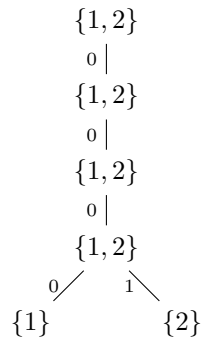
9. We obtain $\mathsf{N}(i) = \{\tau_1, ..., \tau_i\}$ and $\mathsf{B}(i)$, the associated B-C. If $i < S$ increment it by one and repeat all. Otherwise quit.

Example 5.6.3. Consider the set:
$\mathbf{X} = \{(0,0,0,0), (0,0,0,1), (0,1,2,3), (1,0,0,0), (1,0,0,1), (1,1,2,3)\} \subseteq \mathbb{R}^4$.
The first point, $P_1 = (0,0,0,0)$, corresponds to $\tau_1 = 1$:

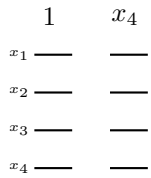
$$
\begin{array}{l}
\{1\} \\
0 \,\big|\, \\
\{1\} \\
0 \,\big|\, \\
\{1\} \\
0 \,\big|\, \\
\{1\} \\
0 \,\big|\, \\
\{1\}
\end{array}
$$



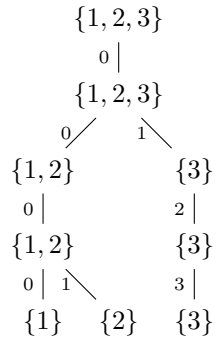
We set $D = \emptyset$ and we proceed with $P_2 = (0, 0, 0, 1)$.



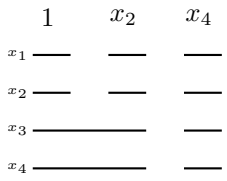
The σ-value is $s = 4$, whereas the B-C antecedent is clearly P_1. Since there is not a x_4-bar after the one over which τ_1 lies, we construct it, setting $\tau_2 = x_4$. The Bar Code turns out to be:



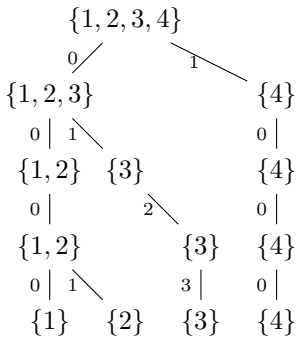
For $P_3 = (0, 1, 2, 3)$ we have



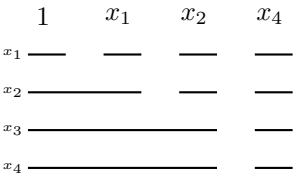
and $s = 2$, so the term τ_3 we have to determine will lie on the first x_3-bar of the first x_4-bar. The B-C antecedent is then P_1 and we construct a new x_2-bar, getting $\tau_3 = x_2$.



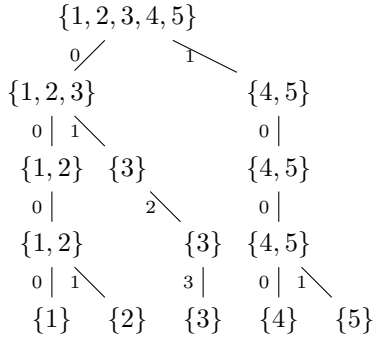
For $P_4 = (1, 0, 0, 0)$, $s = 1$.



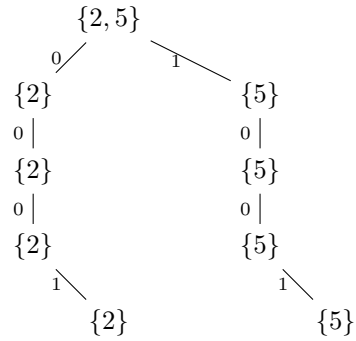
The B-C antecedent is P_1 and we construct a new x_1-bar, so $\tau_4 = x_1$.



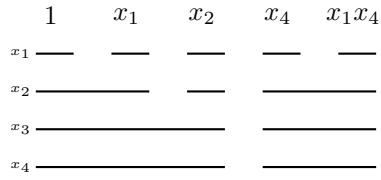
For $P_5 = (1, 0, 0, 1)$, $s = 4$ and $l = 4$:



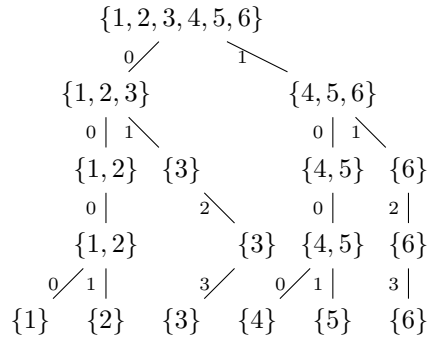
We restrict to the second x_4-bar, setting $D[2] = 1$. This means restricting to the trie



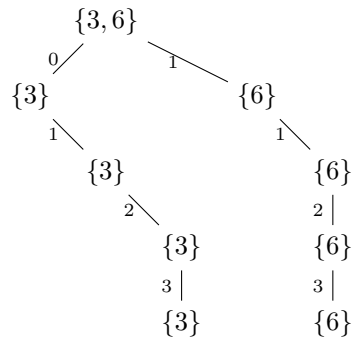
In this recursive step, we have $s = 1$, $l = 2$, then $\tau_5 = x_1x_4$.



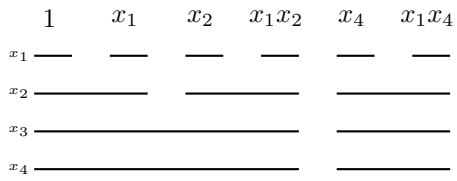
Finally, we deal with $P_6 = (1, 1, 2, 3)$, for which $s = 2$ and $l = 4$.



We restrict to P_3, P_6, so $D[3] = 1$:



and $s = 1$, $l = 3$ then $\tau_6 = x_1x_2$.
The Groebner escalier is $\mathsf{N} = \{1, x_1, x_2, x_1x_2, x_4, x_1x_4\}$ and the Bar Code is



Remark 5.6.4. If $\min(\tau_i) = x_j$, the algorithm adds a new j-bar to the diagram and, by the properties of $P_{x\_}(\_)$, also an l-bar lying over it, for each $l = 1, ..., j-1$. The $(j+1), ..., n$-bars lying under the added ones are simply lengthened by 1 each time, while all the other bars remain unchanged.

The accuracy of the algorithm follows from the one of Cerlienco-Mureddu correspondence, since we are essentially following the same line, exploiting the information we obtain at each step.
The algorithm terminates since it performs a loop on $|\mathbf{X}|$ and reads the trie, whose levels are n and these numbers are finite.
Let us now deal with the computational complexity of the algorithm.
As seen in chapter 2, the complexity of the original Lex Game algorithm is:

$$O(nS + S\min(S, nr)),$$

where S is the number of points in the given finite set $\mathbf{X}$ and n is the number of variables in the ring, i.e. the complexity of the (iterative) construction of the point trie, since the construction of the lex trie is $O(nS)$.

Let us now examine the differences between our algorithm and the Lex Game.

Fix a point P_j, $j \neq 1$. In step j, we first compute the analogous of Cerlienco-Mureddu σ-value s and of the σ-antecedent; we exploit them in order to settle the exponent of the maximal variable in the associated term τ_j.
Such a step is totally equivalent to one of the iterative steps in the point trie construction, so, for each point, we will have $O(n + \min(S, nr))$.
Then we have the inductive step, that is essentially the composition of the following steps:

1. take the s bar containing τ_j (which has been fixed in the first step) and restrict to the corresponding points in the point trie: we get a *reduced point trie* (RPT from now on). This goal is achieved exploiting the list D, whose nonempty entries are only the ones associated to the paths we are restricting to;

2. find the lowest level $f + 1$ in the RPT in which P_j forks, finding the σ-value and the f-antecedent;

This settles the penultimate variable appearing in τ_j. We have to repeat the above two steps for each variable occurring in τ_j.
By the way, we have to point out that if we are in an x_i-bar different from the first (and this is the case for each recursion step), we need to have *at least* half of the $P_1, ..., P_{j-1}$ in the first bar, by the admissibility for Bar-Codes. This means that the RTP we construct contains at most half of the points in the first recursion step, a quarter of the points in the second and so on.
We remind also that *each level is examined twice*.
This leads to the following complexity for P_j: $O(nr + 2\sum_{i=1}^{n} \frac{S}{2^i})$, where r is the maximal number of forks depending on a node. We can conclude that the complexity for a single point is $O(n + \min(S, nr) + nr + S) = O(nr + S + \min(S, nr))$
Now, we have to deal with S points, so we get $O(nrS + S^2 + S\min(S, nr))$.
Clearly $O(nrS + S^2 + S\min(S, nr)) \geq O(nS + S\min(S, nr))$ and it is due to having an iterative construction.
The complexity of our algorithm is strictly inferior than the complexity $O(n^2S^2)$ of the original Cerlienco-Mureddu algorithm, where the main advantage was exactly iterativity.

5.7 The star set and the monomial basis.

As explained in section 5.2, we can associate the star set $\mathcal{F}_{\mathsf{N}}$ to each B-C, corresponding to an order ideal N.
In general, $\mathcal{F}_{\mathsf{N}}$ is not the minimal set of generators for the monomial ideal I whose Groebner

escalier is N. In remark 5.3.4, we showed that $\mathsf{G}(I) \subseteq \mathcal{F}_{\mathsf{N}} \subseteq B(I)$.
First of all, we explain how $\mathcal{F}_{\mathsf{N}}$ is modified by the insertion of a new element in N.
Consider a finite set of distinct points $\mathbf{X}' = \{P_1, ..., P_{S-1}\}$.
Suppose we have found the Groebner escalier of $I(\mathbf{X}')$, namely $\mathsf{N}' = \{\tau_1, ..., \tau_{S-1}\}$ and suppose the Bar-Code to have been drawn.
Moreover, we suppose $\mathcal{F}_{\mathsf{N}'}$ to be known, so we have already associated the star set to the Bar Code.
We add to $\mathbf{X}'$ a new point P_S, obtaining the set $\mathbf{X} = \{P_1, ..., P_S\}$.
We explained before that we can get the term τ_S corresponding to the new point P_S exploiting the Bar Code. We obtain this way the Groebner escalier associated to $I(\mathbf{X})$, namely $\mathsf{N} = \{\tau_1, ..., \tau_S\}$ and we modify consequently the Bar Code.
Let $x_h = \min(\tau_S)$ be the minimal variable appearing in the new term τ_S.
The Bar-Code is modified this way:

- for each x_i, $i \leq h$, we add a new x_i-bar under τ_S;
- for each x_i, $i > h$ we extend the x_i-bar under τ_S.

This implies that we have to modify only the stars lying on lines corresponding to the variables x_i, $i \leq h$.
Since setting a star on the i-th line means looking at the "holes" in the $(i+1)$-th line, we have to look at lines $2, ..., h+1$.
More precisely, we proceed this way.

1 Look at the $(h+1)$-bar lying under τ_S. Since we added a new h-bar, we have to remove the star before the h-bar corresponding to τ_S, replacing it with a star *after* that bar.

2 We add a star after each $1, ..., h-1$ bar lying under τ_S.

We obtain this way the star set $\mathcal{F}_{\mathsf{N}}$.
If we want to obtain the monomial basis $\mathsf{G}(I(\mathbf{X}))$, we only have to check whether the new inserted elements are multiple of the previous terms in the same bar.
The construction above bases on the following

Proposition 5.7.1. Let $\mathsf{N} = \mathsf{N}(I)$, $|\mathsf{N}| < \infty$ be the Groebner escalier associated to a zerodimensional radical ideal and let $\mathcal{F}_{\mathsf{N}}$ be the corresponding (finite) star set.
Given $\tau \in \mathsf{G}(I)$, we denote by $\mathsf{N}' = \mathsf{N}(I) \cup \{\tau\}$ the order ideal obtained by adding τ to N, that is naturally associated to a monomial ideal J, so that $\mathsf{N}' = \mathsf{N}(J)$. It holds:

$$\mathcal{F}_{\mathsf{N}'} = (\mathcal{F}_{\mathsf{N}} \setminus \{\tau\}) \cup \{x_j\tau,\ x_j \leq \min(\tau)\}.$$

Proof: By remark 5.3.4, we have $\mathsf{G}(J) \subseteq \mathcal{F}_{\mathsf{N}'} \subseteq \mathsf{B}(J)$ and, by definition of border set, $\mathsf{B}(J) = (\mathsf{B}(I) \setminus \{\tau\}) \cup \{x_j\tau,\, j = 1, ..., n\}$.
Clearly, if $x_j \leq \min(\tau)$ then $\frac{x_j\tau}{\min(x_j\tau)} = \frac{x_j\tau}{x_j} = \tau \in \mathsf{N}'$.
If $x_j > \min(\tau)$ we have two possibilities:

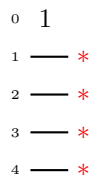
a. $x_j\tau \in \mathcal{F}_{\mathsf{N}}$: in this case $\frac{x_j\tau}{\min(\tau)} \in \mathsf{N}$ and $\frac{x_j\tau}{\min(\tau)} \neq \tau$, so $x_j\tau \in \mathcal{F}_{\mathsf{N}'}$

b. $x_j\tau \notin \mathcal{F}_{\mathsf{N}}$: in this case $\frac{x_j\tau}{\min(\tau)} \neq \tau \notin \mathsf{N}$ and then $x_j\tau \notin \mathcal{F}_{\mathsf{N}'}$.

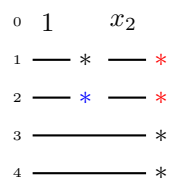
All the terms in $\mathcal{F}_{\mathsf{N}} \setminus \{\tau\}$ also belong to $\mathcal{F}_{\mathsf{N}'}$ whereas, for each $\sigma \in (\mathsf{B}(I) \setminus \{\tau\}) \setminus \mathcal{F}_{\mathsf{N}}$, if $\frac{\sigma}{\min(\sigma)} = \tau$ then we are in case b. above; otherwise $\sigma \notin \mathcal{F}_{\mathsf{N}'}$. $\diamond$

We will see in chapter 6 that the set $\mathcal{F}_{\mathsf{N}}$ represents the leading set for a lexicographical involutive basis. Let us consider an example.

Example 5.7.2. We start with one single point in $\mathbb{R}^4$, namely $P_1 = (0, 0, 0, 0)$. We set $I_1 = I(\{P_1\})$ and $\mathsf{N}_1 = \mathsf{N}(I_1)$.
Applying the Bar-Code algorithm for the Groebner escalier, we get $\mathsf{N}_1 = \{1\}$ and the diagram below:

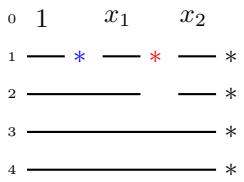


The stars (read from the top to the bottom) correspond to the terms x_1, x_2, x_3, x_4.
We get $\mathcal{F}_{\mathsf{N}_1} = \mathsf{G}(I_1) = \{x_1, x_2, x_3, x_4\}$.
We add a new point $P_2 = (0, 1, 0, 0)$ and we set $I_2 = I(\{P_2\})$ and $\mathsf{N}_2 = \mathsf{N}(I_2)$.
We get $\mathsf{N}_2 = \{1, \tau_2 = x_2\}$, with the Bar-Code below:

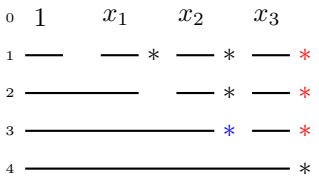


The red stars are the modified ones, whereas the blue one is the one we delete.
We get $\mathcal{F}_{\mathsf{N}_2} = \{x_1, x_1x_2, x_2^2, x_3, x_4\}$ and, being $x \mid xy$, $\mathsf{G}(I_2) = \{x_1, x_2^2, x_3, x_4\}$.

Setting $P_3 = (1,0,0,0)$ and $I_3 = I(\{P_3\})$, we get $\mathsf{N}_3 = \mathsf{N}(I_3) = \{1, \tau_2, \tau_3 = x_1\}$.
The associated Bar-Code is:



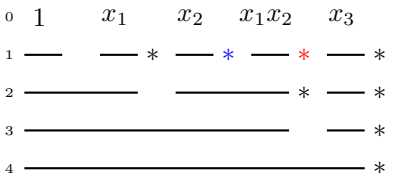
In this case the star set coincides with the monomial basis, having:
$\mathcal{F}_{\mathsf{N}_3} = \mathsf{G}(I_3) = \{x_1^2, x_1x_2, x_2^2, x_3, x_4\}$.
We consider $P_4 = (1,0,1,0)$ and we define: $I_4 = I(\{P_1, P_2, P_3, P_4\})$.
The Groebner escalier is $\mathsf{N}_4 = \mathsf{N}(I_4) = \{1, \tau_2, \tau_3, \tau_4 = x_4\}$.



Removing the blue star and putting on the red one, we get:

$$\mathcal{F}_{\mathsf{N}_4} = \mathsf{G}(I_4) = \{x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_4\}$$

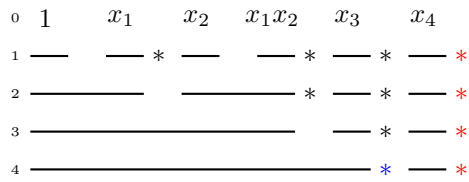
Considered $P_5 = (1,1,0,0)$, we have: $I_5 = I(\{P_1, P_2, P_3, P_4, P_5\})$ and
$\mathsf{N}_5 = \mathsf{N}(I_5) = \{1, \tau_2, \tau_3, \tau_4, \tau_5 = x_1x_2\}$.
The associated Bar-Code is:



Removing the blue star and adding the red one, we get $\mathcal{F}_{\mathsf{N}_5} = \{x_1^2, x_1^2x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_4\}$.
This time, the star set does not coincide with the monomial basis, namely

$$\mathsf{G}(I_5) = \{x_1^2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_4\}.$$

The point $P_6 = (1,0,0,1)$ corresponds to $\tau_6 = t$, so $I_6 = I(\{P_1, P_2, P_3, P_4, P_5, P_6\})$ and
$\mathsf{N}_6 = \mathsf{N}(I_6) = \{1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\}$:



Removing the blue star and adding the red ones, we get

$$\mathcal{F}_{\mathsf{N}_6} = \{x_1^2, x_1^2x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\}.$$

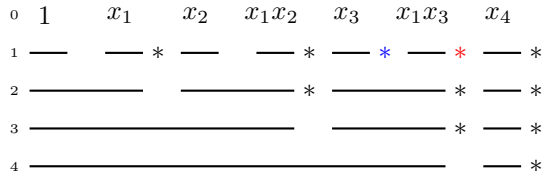
The monomial basis is

$$\mathsf{G}(I_6) = \{x_1^2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\}.$$

We add $P_7 = (0, 0, 1, 0)$ and we have

$I_7 = I(\{P_1, P_2, P_3, P_4, P_5, P_6, P_7\})$, $\mathsf{N}_7 = \mathsf{N}(I_7) = \{1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7 = x_1x_3\}$.

The Bar-Code is:



We obtain

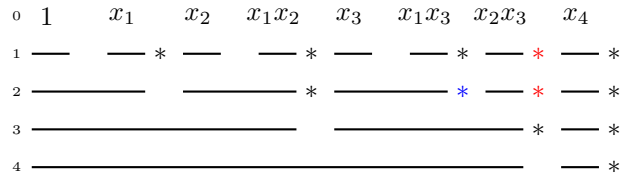
$$\mathcal{F}_{\mathsf{N}_7} = \{x_1^2, x_1^2x_2, x_2^2, (x_1^2x_3), x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\}$$

and

$$\mathsf{G}(I_7) = \{x_1^2, x_2^2, (x_1^2x_3), x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\}.$$

Finally, for $P_8 = (1, 1, 1, 0)$, we set $I_8 = I(\{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\})$ and $\mathsf{N}_7 = \mathsf{N}(I_7) = \{1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7, \tau_8 = x_2x_3\}$.

The Bar-Code is displayed below:



We finally get

$$\mathcal{F}_{\mathsf{N}_8} = \{x_1^2, x_1^2x_2, x_2^2, x_1^2x_3, x_1x_2x_3, x_2^2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\},$$

and

$$\mathsf{G}(I_8) = \{x_1^2, x_2^2, x_1x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\}.$$

Remark 5.7.3. Let B be a B-C and suppose the above steps have been performed. If we read the terms corresponding to the remaining stars proceeding vertically, from the leftmost star to the rightmost one, we obtain the elements of $\mathsf{G}(I)$ *ordered w.r.t* $<$, simply by construction. Indeed, given $\tau_j = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, it is clearly obvious that

$$x_i P_{x_i}(\tau_j) = x_i^{\alpha_i+1} x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} < x_{i+1}^{\alpha_{i+1}+1} x_{i+2}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} = x_{i+1} P_{x_{i+1}}(\tau_j),$$

so the lex inequality holds for terms corresponding to superimposed stars.
Now, let x_h be the maximal variable for which the h-bar underlying τ_j is followed by a star. This means that the last bar breaking after τ_j is the underlying $(h+1)$-bar. We have then to compare $x_1 P_{x_1}(\tau_{j+l})$, $l > 0$ and $x_h P_{x_h}(\tau_j)$. The terms over the subsequent $(h+1)$-bar, w.r.t. τ_j, have either a bigger $(h+1)$-degree or a bigger k-degree, for $k > h+1$. From this fact we can conclude that $x_h P_{x_h}(\tau_j) < x_1 P_{x_1}(\tau_{j+l})$.

5.8 A Bar Code version of the Axis of Evil algorithm.

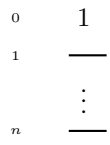
In this section, we develop a third version of the Axis of Evil algorithm. Such a version will be *iterative* on the elements of the given finite set $\mathbf{X} = \{P_1, ..., P_S\}$ of distinct points and it will exploit the Bar Code structure in order to give the Axis of Evil factorization for:

- a lexicographical involutive basis $I(\mathbf{X})$;
- a minimal lexicographical Groebner basis of $I(\mathbf{X})$.

If B is the B-C corresponding to $\mathsf{N} = \mathsf{N}(I(\mathbf{X}))$, we associate a polynomial to each bar in B in such a way that, if $p_j^{(i)}$ is the polynomial associated to a bar $A_j^{(i)}$ in the i-th line, $i = 1, ..., n$; $j = 1, ..., |\mu(i)|$, then $\mathsf{T}(p_j^{(i)}) = x_i$. We also show how to multiply the obtained factors in order to get the factorized bases $\mathcal{J}_S$ and $\mathcal{G}_S$ for $I(\mathbf{X})$.
We give first the main algorithm, supposing the following subroutines to be known:

- $Tp(l, \mathsf{B}, \tau_i)$, which is devoted to the computations of triangular polynomials
- $DiagReading(\mathsf{B}, \tau_i)$ i.e. the one producing the polynomials of the required bases.

Consider initially the case $S = 1$. As explained in section 5.6, the B-C associated to a single point is naturally:



and we then have $\mathsf{N} = \{1\}$.
We define the triangular polynomial $q_1 = 1$ and n sets $\mathfrak{X}_1 = \{x_1 - a_{1,1}\}; ....; \mathfrak{X}_n = \{x_n - a_{1,n}\}$, one for each variable in the polynomial ring. The required polynomials are the elements of $\mathfrak{X} = \bigcup_{i=1}^{n} \mathfrak{X}_i$. We notice that $\mathcal{G}_1 = \mathcal{J}_1 = \{x_1 - a_{1,1}, ..., x_n - a_{1,n}\}$ is the reduced Groebner basis of $I(\{P_1\})$ and it coincides with the involutive basis.
These computations constitute the base step for our algorithm.

Consider now the case $S > 1$. For each $i = 2, ..., S$ perform the following steps.

1. We compute the term $\tau_i = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{N}(i)$, associated to P_i, and $\mathsf{B}(i) = \mathsf{B}_{\mathsf{N}(i)}$, the associated B-C, by the algorithm developed in section 5.6.

2. Compute the triangular polynomial $q_i = Tp(n+1, \mathsf{B}(i), \tau_i)$.

3. As explained in remark 5.6.4, for each $l \in \{1, ..., n\}$, only *one* l-bar, say $A_{j_l}^{(l)}$, is modified by the algorithm of section 5.6. We have to make some small adjustments only on the polynomials corresponding to the modified bars, i.e. $p_{j_1}, ..., p_{j_n}$, maintaining the other ones unchanged. More precisely, if $\min(\tau_i) = x_j, \max(\tau_i) = x_h$, $j, h \leq n$ we proceed as follows:

   (a) for each $l \in \{1, ..., j\}$, compute the polynomial $x_l - a_{i,l}$ and insert it in $\mathfrak{X}_l$ in the position corresponding to the one of the added bar in the l-th line;

   (b) for each $l \in j+1, ..., h$, compute $p_{j_l} - ev_{P_i}(p_{j_l}) Tp(l, \mathsf{B}(i), \tau_i)$;

   (c) for each $l \in h+1, ..., n$, compute $p_{j_l} - ev_{P_i}(p_{j_l}) Tp(n+1, \mathsf{B}(i), \tau_i)$.

4. When $i = S$, if we want to compute the minimal Groebner basis, then compute $\mathsf{G}(I(\mathbf{X}))$ by the algorithm of section 5.7 and, for each $\sigma \in \mathsf{G}(I(\mathbf{X}))$ perform $DiagReading(\mathsf{B}(i), \sigma)$. The elements of $\mathfrak{X}_j$, $j = 1, ..., n$ are the polynomials of theorem 3.4.1, while the output produced by $DiagReading(\mathsf{B}(i), \sigma)$ is a minimal Groebner basis for $I(\mathbf{X})$.
   If the involutive basis is required, we proceed the same way with $\mathcal{F}(I(\mathbf{X}))$ instead of $\mathsf{G}(I(\mathbf{X}))$.

We explain now the procedure $Tp(l, \mathsf{B}, \tau_i)$, which computes the l-th triangular polynomial, for $l = 2, ..., n+1$.

1. For $l = 2, ..., n$ take the l-bar lying under τ_i, say $A_j^{(l)}$ and isolate the block B' composed by the $1, 2, ..., (l-1)$-bars lying over $A_j^{(l)}$. Then, delete $A_j^{(l)}$ and set $\tau_i' = \frac{\tau_i}{x_l^{\alpha_l} \cdots x_n^{\alpha_n}}$.
2. For $l = n+1$ we have $\mathsf{B}' = \mathsf{B}$ and $\tau_i' = \tau_i$.
3. Perform the diagonal reading $DiagReading(\mathsf{B}', \tau_i')$, obtaining a polynomial $f_i \in \mathcal{G}_{i-1}$ such that $T(f_i) = \tau_i'$;
4. Set $q_i = \frac{1}{ev_{P_i}(f_i)} f_i$

Lastly, we examine the procedure $DiagReading(\mathsf{B}, \tau_i)$, whose task is to multiply conveniently the polynomials in $\mathfrak{X}_j$, $j = 1, ..., n$, in order to produce a polynomial $f_i \in \mathcal{G}_{i-1}$ such that $T(f_i) = \tau_i = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$.

1. Compute $f_i^{(n)} = p_1^{(n)} \cdots p_{\alpha_n}^{(n)}$, where $p_1^{(n)}, ..., p_{\alpha_n}^{(n)}$ are the polynomials in $\mathfrak{X}_n$ corresponding to the bars $A_1^{(n)}, ..., A_{\alpha_n}^{(n)}$.
2. Let $A_l^{(n-1)}$ the leftmost bar lying over $A_{\alpha_n}^{(n+1)}$. Then $f_i^{(n-1)} = p_l^{(n-1)} \cdots p_{\alpha_{n-1}}^{(n-1)}$, where $p_l^{(n-1)}, ..., p_{\alpha_{n-1}}^{(n-1)}$ are the polynomials in $\mathfrak{X}_{n-1}$ corresponding to the bars $A_l^{(n-1)}, ..., A_{\alpha_{n-1}}^{(n-1)}$.
3. Repeat step (2) for $n-2, n-3, ..., 1$.
4. $f_i = f_i^{(n)} \cdot f_i^{(n-1)} \cdots f_i^{(1)}$.

Remark 5.8.1. The subroutine $Tp(l, \mathsf{B}, \tau_i)$ produces interpolators à la Moeller. It essentially computes the polynomial of the minimal Groebner basis $\mathcal{G}_{i-1}$, whose leading term is τ_i, without computing or storing the whole $\mathcal{G}_{i-1}$. Thanks to the B-C structure and to the procedure $DiagReading(\mathsf{B}, \tau_i)$, we can exploit the (previously computed) polynomials of $\mathfrak{X}_j$, $j = 1, ..., n$ in order to get the required interpolators.

The algorithm explained above ensures the existence of the polynomials of the form stated in theorem 3.4.1.

We prove now the following

Proposition 5.8.2. With the above notation, we have

$$I := (\{DiagReading(\mathsf{B}, \sigma) \,|\, \sigma \in \mathsf{G}(I(\mathbf{X}))\}) = I(\mathbf{X}).$$

Proof: Consider the polynomial γ_τ associated to the term $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in \mathsf{G}(I)$.
We prove that it vanishes on $P_\mu \in \mathbf{X}$, corresponding to the term $\mu = x_1^{\beta_1} \cdots x_n^{\beta_n} \in \mathsf{N}(I)$.
Since $\tau \in \mathsf{G}(I)$ and $\mu \in \mathsf{N}(I)$, $\tau \neq \mu$. Therefore, there are only two possibilities:

1) $\mu <_{Lex} \tau$. In this case the polynomial obviously vanishes by *DiagReading*, since we pick a bar under μ and the polynomial corresponding to that bar has already been interpolated at the point.

2) $\mu >_{Lex} \tau$. This time $\exists i \in \{1, ..., n\}$ such that $\beta_i > \alpha_i$, $\beta_j = \alpha_j$ for each $j \in \{i+1, ..., n\}$.
By Cerlienco-Mureddu correspondence, $\exists \mu' := x_1^{\beta'_1} \cdots x_n^{\beta'_n} \in \mathsf{N}(I)$ such that:

a. $\Phi^{-1}(\mu') = P_{\mu'}$ with $\pi_{i-1}(P_\mu) = \pi_{i-1}(P_{\mu'})$;

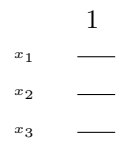
b. $\beta'_h = \alpha_h, \forall h \in \{i, i+1, ..., n\}$.

If $\mu' < \tau$, then, as in 1, γ_τ vanishes in $P_{\mu'}$ and the linear factor making our polynomial vanish in $P_{\mu'}$ is computed using at most the first $i-1$ coordinates of P_μ, so that γ_τ turns out to vanish also in P_μ.
If $\mu' > \tau$, we can repeat with μ' instead of μ and conclude by induction. $\diamond$

Example 5.8.3. Let us consider the set

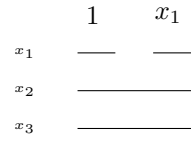
$$\mathbf{X} = \{(0,0,0), (1,2,3), (1,4,5), (0,1,4), (1,4,6), (0,0,2), (0,2,2)\}.$$

Take first $P_1 = (0,0,0)$, for which $\tau_1 = 1$, then $\mathsf{N}(1) = \{1\}$ and $\mathsf{B}(1)$ is the B-C displayed below.



We have $q_1 = 1$ and $\mathfrak{X}_1 = \{x_1\}$, $\mathfrak{X}_2 = \{x_2\}$, $\mathfrak{X}_3 = \{x_3\}$.

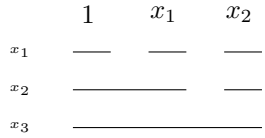
Consider then $P_2 = (1,2,3)$, for which $\tau_2 = x_1$, $\mathsf{N}(2) = \{1, x_1\}$ and $\mathsf{B}(2)$ is



The diagonal reading is trivial and it leads to $q_2 = x_1$. The factors are:
$\mathfrak{X}_1 = \{x_1, x_1 - 1\}$ to get this set from the $\mathfrak{X}_1$ of the previous step, we add the polynomial

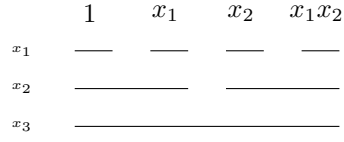
corresponding to the new bar;
$\mathfrak{X}_2 = \{x_2 - 2x_1\}$, obtained as $x_2 - ev_{P_2}(x_2)q_2$;
$\mathfrak{X}_3 = \{x_3 - 3x_1\}$, i.e. $x_3 - ev_{P_2}(x_3)q_2$.

Take then $P_3 = (1, 4, 5)$. We get $\tau_2 = x_2$, $\mathsf{N}(3) = \{1, x_1, x_2\}$ and $q_3 = \frac{1}{2}(x_2 - 2x_1)$.



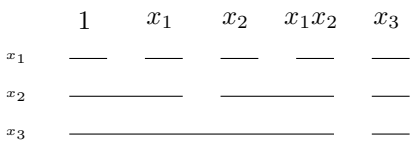
The factors are:
$\mathfrak{X}_1 = \{x_1, x_1 - 1, x_1 - 1\}$
$\mathfrak{X}_2 = \{x_2 - 2x_1, x_2 - 4\}$
$\mathfrak{X}_3 = \{x_3 - x_2 - x_1\}$ i.e. $(x_3 - 3x_1) - ev_{P_3}(x_3 - 3x_1)q_3$.

Consider $P_4 = (0, 1, 4)$, which is associated to $\tau_4 = x_1x_2$. The current Groebner escalier is $\mathsf{N}(4) = \{1, x_1, x_2, x_1x_2\}$, corresponding to the following $\mathsf{B}(4)$:



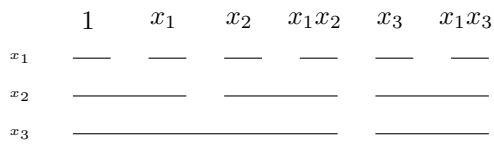
The polynomials in the three variables x_1, x_2, x_3 are:
$\mathfrak{X}_1 = \{x_1, x_1 - 1, x_1 - 1, x_1\}$;
$\mathfrak{X}_2 = \{x_2 - 2x_1, x_2 - 4 - 3(x_1 - 1)\}$, since $T(q_4) = x_1x_2 > x_2$, we compute $Tp(2, \mathsf{B}(4), \tau_4) = -(x_1 - 1)$;
$\mathfrak{X}_3 = \{x_3 - x_2 - x_1 + 3(x_2 - 2x_1)(x_1 - 1)\}$; being $T(q_4) = x_1x_2 < x_3$ we do not need to compute another interpolator.

Take then $P_5 = (1, 4, 6)$, getting $\tau_5 = x_3$, $\mathsf{N}(5) = \{1, x_1, x_2, x_1x_2, x_3\}$ and $q_5 = x_3 - x_2 - x_1 + 3(x_2 - 2x_1)(x_1 - 1)$.



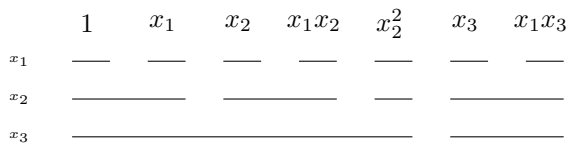
The factors are:
$\mathfrak{X}_1 = \{x_1, x_1 - 1, x_1 - 1, x_1, x_1 - 1\}$
$\mathfrak{X}_2 = \{x_2 - 2x_1, x_2 - 4 - 3(x_1 - 1), x_2 - 4\}$
$\mathfrak{X}_3 = \{x_3 - x_2 - x_1 + 3(x_2 - 2x_1)(x_1 - 1), x_3 - 6\}$.

Take $P_6 = (0,0,2)$, associated to $\tau_6 = x_1x_3$. The current Groebner escalier is $\mathsf{N}(6) = \{1, x_1, x_2, x_1x_2, x_3, x_1x_3\}$ and the interpolator is $q_6 = -\frac{1}{2}(x_1-1)(x_3 - x_2 - x_1 + 3(x_2 - 2x_1)(x_1-1))$.



The factors are:
$\mathfrak{X}_1 = \{x_1, x_1-1, x_1-1, x_1, x_1-1, x_1\}$
$\mathfrak{X}_2 = \{x_2 - 2x_1, x_2 - 4 - 3(x_1-1), x_2 - 4 - 4(x_1-1)\}$
$\mathfrak{X}_3 = \{x_3 - x_2 - x_1 + 3(x_2-2x_1)(x_1-1), x_3 - 6 - 4(x_1-1)\}$; being $T(q_6) = x_1x_3 > x_3$, we compute $q = -(x_1-1)$, via the procedure $Tp(3, \mathsf{B}(6), x_1x_3)$, so restricting to the block containing only $A_5^{(1)}, A_6^{(1)}, A_3^{(2)}$.

The last point, $P_7 = (0,2,2)$ is associated to $\tau_7 = x_2^2$, so the final Groebner escalier is $\mathsf{N} = \mathsf{N}(7) = \{1, x_1, x_2, x_2^2, x_1x_2, x_3, x_1x_3\}$. We have $q_7 = \frac{1}{2}(x_2 - 2x_1)(x_2 - 4 - 3(x_1-1))$. We compute now the minimal monomial basis $\mathsf{G} = \{x_1^2, x_1x_2^2, x_2^3, x_2x_3, x_3^2\}$ and the set $\mathcal{F}(I) = \{x_1^2, x_1^2x_2, x_1x_2^2, x_2^3, x_1^2x_3, x_2x_3, x_3^2\}$.



The factors are:
$\mathfrak{X}_1 = \{x_1, x_1-1, x_1-1, x_1, x_1, x_1-1, x_1\}$
$\mathfrak{X}_2 = \{x_2 - 2x_1, x_2 - 4 - 3(x_1-1), x_2 - 2, x_2 - 4 - 4(x_1-1)\}$
$\mathfrak{X}_3 = \{x_3 - x_2 - x_1 + 3(x_2-2x_1)(x_1-1) + 3(x_2-2x_1)(x_2-4-3(x_1-1)), x_3 - 6 - 4(x_1-1)\}$.
At the end we have

$$\mathcal{G}_7 = \{x_1(x_1-1), x_1(x_2-2x_1)(x_2-4-3(x_1-1)), (x_2-2x_1)(x_2-4-3(x_1-1))(x_2-2),$$

$$(x_2-4-4(x_1-1))(x_3-x_2-x_1+3(x_2-2x_1)(x_1-1)+3(x_2-2x_1)(x_2-4-3(x_1-1))),$$

$$(x_3-6-4(x_1-1))(x_3-x_2-x_1+3(x_2-2x_1)(x_1-1)+3(x_2-2x_1)(x_2-4-3(x_1-1)))\}$$

and

$$\mathcal{J}_7 = \{x_1(x_1-1), (x_2-2x_1)x_1(x_1-1), x_1(x_2-2x_1)(x_2-4-3(x_1-1)),$$
$$(x_2-2x_1)(x_2-4-3(x_1-1))(x_2-2),$$

$$x_1(x_1-1)(x_3-x_2-x_1+3(x_2-2x_1)(x_1-1)+3(x_2-2x_1)(x_2-4-3(x_1-1))),$$

$$(x_2-4-4(x_1-1))(x_3-x_2-x_1+3(x_2-2x_1)(x_1-1)+3(x_2-2x_1)(x_2-4-3(x_1-1))),$$

$$(x_3-6-4(x_1-1))(x_3-x_2-x_1+3(x_2-2x_1)(x_1-1)+3(x_2-2x_1)(x_2-4-3(x_1-1)))\}$$

5.9 Enumerative combinatorics on strongly stable ideals.

This section is about a possible application of Bar Codes to enumerative combinatorics. Using the Bar Code structure, we want to approach the quest for an integer bounding the number of some special zerodimensional monomial ideals, called *strongly stable ideals*, with fixed constant Hilbert polynomial.
We will start to outline a connection between two objects, which appear to be very different and far, namely:

a) strongly stable monomial ideals $I \lhd \mathcal{P}$;

b) integer partitions and plane partitions.

Objects of type a) belong to the field of commutative algebra, whereas those of type b) are related to enumerative combinatorics. Linking them by means of the Bar Code structure of the Groebner escaliers, we will give a bound to the number of zerodimensional strongly stable monomial ideals of a fixed multiplicity.
First of all, we recall the definition of strongly stable ideal. Chapter 6 will deal with strongly stable ideals.

Definition 5.9.1 ([27]). A monomial ideal $I \lhd \mathcal{P} = \mathbf{k}[x_1, ..., x_n]$ is called *strongly stable* if, for every term $\tau \in I$ and pair of variables $x_i,\ x_j$ such that $x_i|\tau$ and $x_i < x_j$, then also $\frac{\tau x_j}{x_i}$ belongs to I or, equivalently, for every $\sigma \in \mathsf{N}(I)$, and pair of variables $x_i,\ x_j$ such that $x_i|\sigma$ and $x_i > x_j$, then also $\frac{\sigma x_j}{x_i}$ belongs to $\mathsf{N}(I)$.

A first property, useful for the following computations, is that Bar Codes of strongly stable ideals are *full*.

Lemma 5.9.2. For all strongly stable ideal $J \lhd \mathcal{P}$, it holds:

$$\mathcal{F}(J) = \{x^\alpha \in \mathcal{T} \setminus \mathsf{N}(J) \,|\, \frac{x^\alpha}{min(x^\alpha)} \in \mathsf{N}(J)\} = \mathsf{G}(J),$$

i.e. all the stars in the associated Bar Code correspond to a term of the monomial basis.

Proof: The inclusion $\mathsf{G}(J) \subseteq \mathcal{F}(J)$ holds for any monomial ideal $I \triangleleft \mathbf{k}[x_1, ..., x_n]$ (5.3.4), so we only prove the other one. Actually, it easily comes from the definition of strongly stable ideal. Indeed, consider $x^\alpha \in \mathcal{F}(J)$. We show that all its predecessors belong to the Groebner escalier $\mathsf{N}(J)$.
Let $x_i = \min(x^\alpha)$ and let $x_j > x_i$ be a variable appearing in x^α with nonzero exponent.
By definition $\frac{x^\alpha}{x_i} \in \mathsf{N}(J)$ and also $\frac{x^\alpha}{x_j} = \frac{x^\alpha}{x_i}\frac{x_i}{x_j} \in \mathsf{N}(J)$, so we can conclude. $\diamond$

We will see another proof of this fact in chapter 6, while defining stable ideals.
Let us now examine the shape of the Bar Code of a strongly stable ideal, that for short we will call *strongly stable Bar Code*.

Proposition 5.9.3. Let $J \triangleleft \mathbf{k}[x_1, ..., x_n]$ be a zerodimensional strongly stable monomial ideal, and $\mathsf{B} := \mathsf{B}_{\mathsf{N}(J)}$ the Bar Code associated to its (finite) Groebner escalier.
Fixed a $(i+2)$-bar A, for $i \in \{1, ..., n-1\}$[5], let $C_1, ..., C_h$ be the $(i+1)$-bars over A. Then $l_i(C_1) > ... > l_i(C_h)$.

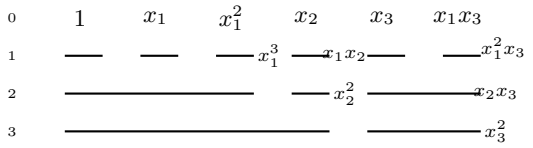
Proof: In order to prove the assertion, we proceed by contradiction.
Since the case $l_i(C_j) < l_i(C_l)$ for $i < l$ implies that the Bar Code is even not admissible, suppose that $l_i(C_j) = l_i(C_l)$ for $i < l$ and take $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, i.e. the rightmost term lying over C_l. Over C_l we have $\alpha_i + 1$ i-bars.
By definition of strongly stable ideal, the term $\sigma = x_1^{\alpha_1} \cdots x_i^{\alpha_i+1} x_{i+1}^{\alpha i+1-1} \cdots x_n^{\alpha_n} \in \mathsf{N}(J)$. But this implies that we should have at least $\alpha_i + 2 > \alpha_i + 1$ i-bars over C_l and this is a contradiction. $\diamond$

Remark 5.9.4. The condition of proposition 5.9.3 holds for each strongly stable ideal, but there are also *non-strongly stable* monomial ideals fulfilling them, so the reverse does not hold.

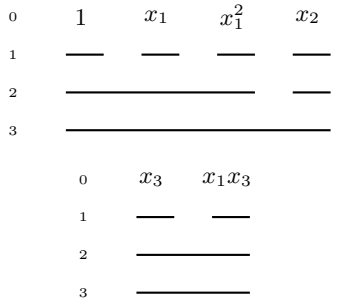
Let us see an example of the problem emphasized in remark 5.9.4.

Example 5.9.5. Let $J = (x_1^3, x_1x_2, x_2^2, x_1^2x_3, x_2x_3, x_3^2) \triangleleft \mathbf{k}[x_1, x_2, x_3]$ (see example 5.3.5).
The corresponding Groebner escalier is $\mathsf{N} = \mathsf{N}(J) = \{1, x_1, x_1^2, x_2, x_3, x_1x_3\}$ and the associated Bar Code B is displayed in the picture below:

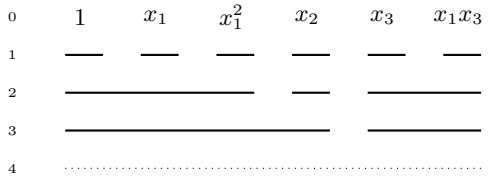
[5] For $i = n-1$, we consider as $(n+1)$-bar a line underlining the whole diagram. We use it only in theory, for some proofs, even if we *never* draw it concretely.



The bar list of B is $(6, 3, 2)$ and the star set is $\mathcal{F}_{\mathsf{N}} = \mathsf{G}(J) = \{x_1^3, x_1x_2, x_2^2, x_1^2x_3, x_2x_3, x_3^2\}$.
The condition of Lemma 5.9.3 holds with $i = 1, 2$. Indeed we can only isolate the sub-Bar Codes



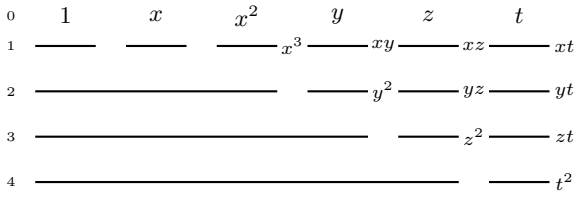
for which the condition holds and the same is valid for the x_2-bars, with respect to the whole diagram:



Anyway, J is not a strongly stable ideal, since $xz \in \mathsf{N}(J)$, while $xy \in J$.

Remark 5.9.6. Let B be the Bar Code associated to the Groebner escalier $\mathsf{N}(J)$ of a zerodimensional strongly stable ideal J.
If we consider the bars $A_1^{(i+1)}, ..., A_{\mu(i+1)}^{(i+1)}$ in the $(i+1)$-th line, it is not true in general that $l_i(A_1^{(i+1)}) > ... > l_i(A_{\mu(i+1)}^{(i+1)})$, as shown in the examples below.

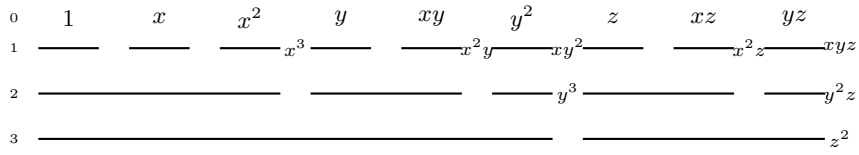
Example 5.9.7. For $I = (t^2, tz, z^2, ty, zy, y^2, tx, zx, yx, x^3) \lhd \mathbf{k}[x, y, z, t]$ the associated B-C is:



The ideal I is strongly stable, but we have

$$2 = l_2(A_1^{(3)}) > l_2(A_2^{(3)}) = l_2(A_3^{(3)}) = 1.$$

Example 5.9.8. The monomial ideal $I = (z^2, zy^2, y^3, zyx, y^2x, zx^2, yx^2, x^3) \triangleleft \mathbf{k}[x, y, z]$ is associated to the Bar Code displayed below

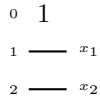


This monomial ideal is strongly stable, but

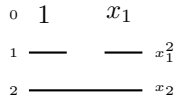
$$l_1(A_1^{(2)}) = 3,\ l_1(A_2^{(2)}) = 2,\ l_1(A_3^{(2)}) = 1,\ l_1(A_4^{(2)}) = 2 \text{ and } l_1(A_5^{(2)}) = 1,$$

so *the considered lengths are not all in nonincreasing order*.

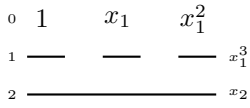
Let us start examining the Bar Code structure of the Groebner esalier for zerodimensional strongly stable ideals, starting from the case of two variables.
First of all, let us look to some examples.
The only strongly stable ideal with affine Hilbert polynomial equal to 1 is the maximal ideal $J_1 = (x_1, x_2)$, whose Bar Code is trivially



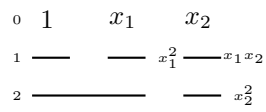
The associated bar list is then $(1, 1)$.
If we examine the strongly stable monomial ideals in two variables with affine Hilbert polynomial equal to 2 we get $J_1 = (x_1^2, x_2)$, whose Bar Code is



and the associated bar list is $(2, 1)$.
For the affine Hilbert polynomial $H\_(t) = 3$ we have $J_1 = (x_1^3, x_2)$ $J_2 = (x_1^2, x_1x_2, x_2^2)$.
The Bar Code associated to J_1 is



and the bar list is $(3, 1)$.
The Bar Code associated to J_2 is



and the bar list is $(3, 2)$.
We summarize in the following table the bar lists of strongly stable ideals corresponding to each $H\_(t)$.

$H\_(t)$	Bar lists	Ideals
$H\_(t) = 1$	$(1, 1)$	(x_1, x_2)
$H\_(t) = 2$	$(2, 1)$	(x_1^2, x_2)
$H\_(t) = 3$	$(3, 1), (3, 2)$	$(x_1^3, x_2), (x_1^2, x_1x_2, x_2^2)$
$H\_(t) = 4$	$(4, 1), (4, 2)$	$(x_1^4, x_2), (x_1^3, x_1x_2, x_2^2)$
$H\_(t) = 5$	$(5, 1), (5, 2), (5, 2)$	$(x_1^5, x_2), (x_1^4, x_1x_2, x_2^2), (x_1^3, x_1^2x_2, x_2^2)$
$H\_(t) = 6$	$(6, 1), (6, 2), (6, 2), (6, 3)$	$(x_1^6, x_2), (x_1^5, x_1x_2, x_2^2), (x_1^4, x_1^2x_2, x_2^2), (x_1^3, x_1^2x_2, x_1x_2^2, x_2)$
$\dots$	$\dots$	$\dots$

Observing the second column of the table, we can notice some "pattern" in their distributions.
Driven by this pattern, we examine more deeply the Bar code structure of these ideals.

For this purpose, we need the following

Definition 5.9.9 ([101])**.** A *partition* of $p \in \mathbb{N}$ is a sequence $(\alpha_1, ..., \alpha_k) \in \mathbb{N}^k$ such that $\sum_{i=1}^{k} \alpha_i = p$ and $\alpha_1 \geq ... \geq \alpha_k$

We regard two partitions as identical if they only differ in the number of terminal 0's. For example $(3, 2, 1) = (3, 2, 1, 0, 0)$.
Informally, we regard a partition $\alpha = (\alpha_1, ..., \alpha_k)$, say with $\alpha_k > 0$ as a way of writing p as a sum of positive integers, disregarding the order of the summands.
The nonzero terms are called *parts* of α and we say that α has k parts if

$$k = |\{i,\, \alpha_i > 0\}|.$$

We are interested to the special case $\alpha_1 > ... > \alpha_k$ i.e. to integer partitions of p into k *distinct parts.*
We are now ready to prove the following proposition

Proposition 5.9.10. The number of strongly stable Bar Codes for terms in $\mathbf{k}[x_1, x_2]$, whose bar list is (p, h), $p, h \in \mathbb{N}$, $p \geq h$ equals the number of integer partitions of p in h distinct parts, namely

$$p = \alpha_1 + ... + \alpha_h,\ \alpha_i > 0,\ i = 1, ..., h.$$

Proof: In order to prove the assertion, we want to establish a biunivocal correspondence between

$$\mathsf{B}_{(p,h)} := \{\text{strongly stable Bar Codes with bar list } (p, h)\}$$

and

$$I_{(p,h)} := \{(\alpha_1, ..., \alpha_h) \in \mathbb{N}^h,\ \alpha_1 > ... > \alpha_h,\ \sum_{i=1}^{h} \alpha_i = p\}.$$

We set then

$$\Xi : \mathsf{B}_{(p,h)} \longrightarrow I_{(p,h)}$$

$$\mathsf{B} \mapsto (l_1(A_1^{(2)}), ..., l_1(A_h^{(2)})).$$

Let B be a strongly stable Bar-Code, whose bar list is (p, h). It is associated to the set of terms $M_{\mathsf{B}} = \mathsf{N}(J)$, for $J \lhd \mathbf{k}[x_1, x_2]$ strongly stable.
The sequence $(l_1(A_1^{(2)}), ..., l_1(A_h^{(2)}))$ satisfies $l_1(A_1^{(2)}) > ... > l_1(A_h^{(2)})$ by proposition 5.9.3 and since $|\mathsf{N}(J)| = p$, then $\sum_{i=1}^{h} l_1(A_i^{(2)}) = p$, so we exactly have an integer partition of p into h distinct parts.
On the other hand, let $(\alpha_1, ..., \alpha_h) \in I_{(p,h)}$. We construct the (unique) Bar Code $\mathsf{B}_{(p,h)}$ associated to this h-tuple, namely a Bar Code formed by h x_2-bars such that over the first x_2-bar there lie α_1 x_1-bars, and so on.
We have to prove that the associated M_{B} is the Groebner escalier $\mathsf{N}(J)$ of a strongly stable ideal $J \lhd \mathbf{k}[x_1, x_2]$.
Consider $\sigma := x_1^{\beta_1} x_2^{\beta_2} \in M_{\mathsf{B}}$, $0 < \beta_2 < h$[6].
By definition 5.9.1, we only have to prove that $\sigma' = \frac{\sigma x_1}{x_2} \in M_{\mathsf{B}}$, but this is obviously true, since, over the $(\beta_2 - 1)$-th x_2-bar it lies at least one x_1-bar more than the x_1-bars lying over the β_2-th x_2-bar. $\diamond$

We point out that, if $H\_(t) = p$, the Bar list $(p, 1)$ corresponds to the ideal $J = (x_1^p, x_2)$ which is a very particular strongly stable ideal: a *lex segment ideal*.
More precisely, for each degree i, J is k-spanned by the first $H\_(i)$ terms w.r.t. lex.
The bar list $(p, 1)$ is clearly the one presenting the minimal value for h. Now we should try to understand which is the maximal value for h.

[6]For $\beta_2 = 0$ there is nothing to prove since we cannot perform any operations as in definition 5.9.1.

Proposition 5.9.11. The maximal value for h in a bar list (p, h) of a strongly stable Bar Code is the maximal integer h such that $\frac{h(h+1)}{2} \leq p$.

Proof: By proposition 5.9.10, the strongly stable Bar Codes are in biunivocal correspondence with the integer partitions of $H\_(t) = p$ into h distinct parts $\alpha_1 + ... + \alpha_h = p$. The minimal values we can assign to α_i, $i = 1, ..., h$ are $(h-1), (h-2), ..., 2, 1$, whose sum is $\frac{h(h+1)}{2}$. Since we are looking for partitions of p, we should have $\frac{h(h+1)}{2} \leq p$. $\diamond$

In order to deal with strongly stable ideals $J \triangleleft \mathbf{k}[x_1, ..., x_n]$ for $n > 2$, the following corollary will be rather useful.

Corollary 5.9.12. The number of strongly stable Bar Codes for terms in $\mathbf{k}[x_1, ..., x_n]$, $n > 2$, whose bar list is $(p, h, 1, ..., 1)$, $p, h \in \mathbb{N}$, $p \geq h$ equals the number of integer partitions of p in h distinct parts, namely

$$p = \alpha_1 + ... + \alpha_h,\ \alpha_i > 0,\ i = 1, ..., h.$$

Moreover, the maximal value for h in the bar list $(p, h, 1, ..., 1)$ is the maximal integer h such that $\frac{h(h+1)}{2} \leq p$.

Proof: It is a straightforward consequence of propositions 5.9.10 and 5.9.11, noticing that, if we have only 1 $x_3, ..., x_n$-bars, $x_3, ..., x_n$ do not occur in any term of M_{B} with nonzero exponent. $\diamond$

By the previous comments, we are able to count the number of strongly stable ideals $J \triangleleft \mathbf{k}[x_1, x_2]$ with $H\_(t, J) = p$.
The following proposition is a consequence of 5.9.10 and 5.9.11.

Proposition 5.9.13. The number of strongly stable ideals J with $H\_(t, J) = p$ is

$$\sum_{i=1}^{h} Q(p, i),$$

where h is the maximal integer such that $\frac{h(h+1)}{2} \leq p$ and $Q(p, i)$ is the number of integer partitions of p into i distinct parts.

The number $Q(p, i)$ of integer partitions of p into i distinct parts has already been studied in literature. For example, we can find in [101] the formulas regulating it:

$$\forall p, i \in \mathbb{N},\ i \neq 1,\ Q(p, i) = P\left(p - \binom{i}{2}, i\right),\ Q(p, 1) = 1$$

where

$$\forall n, k \in \mathbb{N},\ P(n,k) = P(n-1,k-1) + P(n-k,k),$$

with

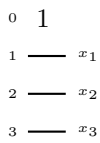
$$\begin{cases} P(n,k) = 0 \text{ for } k > n \\ P(n,n) = 1 \\ P(n,0) = 0 \end{cases}$$

Example 5.9.14. For the polynomial ring $\mathbf{k}[x_1, x_2]$, consider $H\_(t) = 10$.
By our formulas, we have exactly 10 strongly stable monomial ideals with $H\_(t) = 10$.
More precisely, they are:

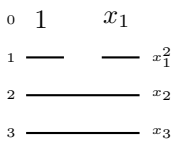
- $J_1 = (x_1^{10}, x_2)$;
- $J_2 = (x_1^9, x_1x_2, x_2^2)$;
- $J_3 = (x_1^8, x_1^2x_2, x_2^2)$;
- $J_4 = (x_1^7, x_1^3x_2, x_2^2)$;
- $J_5 = (x_1^7, x_1x_2^2, x_2x_1^2, x_2^3)$;
- $J_6 = (x_1^6, x_1^4x_2, x_2^2)$;
- $J_7 = (x_1^6, x_1x_2^2, x_1^3x_2, x_2^3)$;
- $J_8 = (x_1^5, x_2^2x_1, x_2x_1^4, x_2^3)$;
- $J_9 = (x_1^5, x_2^2x_1^2, x_2x_1^3, x_2^3)$;
- $J_{10} = (x_1^4, x_2^3x_1, x_2^2x_1^2, x_2x_1^3, x_2^4)$.

Example 5.9.15. The strongly stable monomial ideals with $H\_(t) = 100$ are exactly 444793.

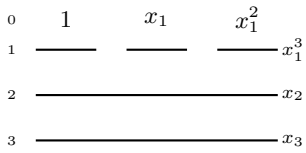
We now try to study the case of three variables, which is a little more cumbersome than the previous case of only two variables.
Let us start with some examples.
If, in $\mathbf{k}[x_1, x_2, x_3]$, $x_1 < x_2 < x_3$, we consider $H\_(t) = 1$, we can associate to it only one strongly stable monomial ideal, namely the maximal ideal $J_1 = (x_1, x_2, x_3)$, whose Bar Code is



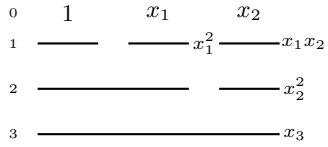
and the associated bar list is $(1, 1, 1)$.
For $H_{\_}(t) = 2$, we get $J_1 = (x_1^2, x_2, x_3) \triangleleft \mathbf{k}[x_1, x_2, x_3]$, whose Bar Code is



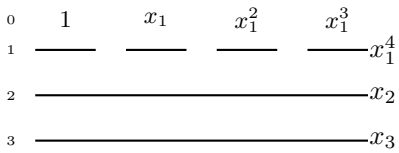
and the corresponding bar list is $(2, 1, 1)$.
Let us take now $H_{\_}(t) = 3$.
The associated strongly stable ideals are $J_1 = (x_1^3, x_2, x_3)$, $J_2 = (x_1^2, x_1x_2, x_2^2, x_3)$, whose bar lists are $(3, 1, 1), (3, 2, 1)$, since their Bar Codes are, respectively,

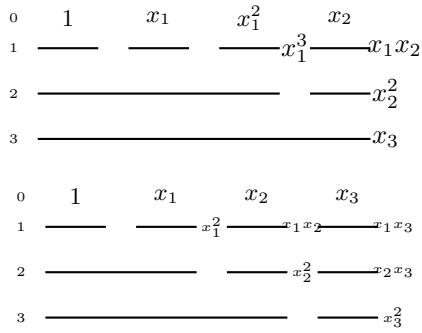


and



If we continue taking $p(t) = 4$ we obtain 3 different strongly stable ideals, namely $J_1 = (x_3, x_2, x_1^4)$, $J_2 = (x_3, x_2^2, x_2x_1, x_1^3)$ and $J_3 = (x_3^2, x_3x_2, x_2^2, x_3x_1, x_2x_1, x_1^2)$.
Their Bar-lists are $(4, 1, 1), (4, 2, 1), (4, 3, 2)$, corresponding to the following Bar Codes:





As for the bidimensional case, we summarize some partial result in the following table:

$H\_(t)$	Bar lists	Ideals
1	$(1,1,1)$	(x_1, x_2, x_3)
2	$(2,1,1)$	(x_1^2, x_2, x_3)
3	$(3,1,1), (3,2,1)$	(x_1^3, x_2, x_3), $(x_1^2, x_1x_2, x_2^2, x_3)$
4	$(4,1,1), (4,2,1), (4,3,2)$	(x_3, x_2, x_1^4), $(x_3, x_2^2, x_2x_1, x_1^3)$, $(x_3^2, x_3x_2, x_2^2, x_3x_1, x_2x_1, x_1^2)$.
5	$(5,1,1), (5,2,1), (5,2,1), (5,3,2)$	$(x_3, x_2, x_1^5), (x_3, x_2^2, x_1x_2, x_1^4)$ $(x_3, x_2^2, x_2x_1^2, x_1^3), (x_3^2, x_3x_2, x_2^2, x_3x_1, x_2x_1, x_1^3)$

Table 5.1: Strongly stable ideals, with affine Hilbert polynomial and bar lists.

By corollary 5.9.12, we can use the formulas for two variables in order to count the strongly stable monomial ideals associated to bar lists of the form $(p, h, 1)$. This means that we only have to deal with the bar lists of the form (p, h, k), such that $k > 1$.

Definition 5.9.16. The *minimal sum* of a given list of positive integers $[\alpha_1, ..., \alpha_g]$ is the integer

$$Sm([\alpha_1, ..., \alpha_g]) := \sum_{i=1}^{g} \frac{\alpha_i(\alpha_i + 1)}{2}.$$

The following lemma is a straightforward consequence of proposition 5.9.3.

Lemma 5.9.17. With the previous notation, it holds:

1. $\min(k) = 2$;

2. $\max(k) = \max_{k\geq 2}\{k|\, \exists L \in I_{(p,k')}$, with $Sm(L) \leq p\}$, $k' = \frac{k(k+1)}{2}$
3. $\min(h) = \frac{k(k+1)}{2}$;
4. $\max(h) = \max_{\frac{k(k+1)}{2}\leq l\leq p-1}\{l|\, Q(h,k) \neq 0$ and $L \in I_{(h,k)} \Rightarrow Sm(L) \leq p\}$.

Thanks to the previous lemma 5.9.17 we know which are the bar lists that occur in the computation for $H\_(t) = p$.
Next step is to understand *how many strongly stable ideals with* $H\_(t) = p$ *and bar-list* (p,h,k) there exist.
More precisely, fixed (p,h,k), we compute the integer partitions of h in k parts, representing the numbers of x_2-bars over the k x_3-bars. Suppose $(\alpha_1, ..., \alpha_k)$, $\alpha_1 > ... > \alpha_k$, $\sum_{i=1}^{k}\alpha_i = h$ being one of these partitions. Then, we construct a $k \times \alpha_1$ matrix M having the following shape:

$$M = \begin{pmatrix} a_{1,1} & ... & ... & ... & ... & ... & a_{1,\alpha_1} \\ 0 & ... & a_{2,2} & ... & ... & a_{2,\alpha_2+1} & 0... \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & ... & 0 & a_{k,k} & ... & a_{k,\alpha_k+k-1} & 0... \end{pmatrix}.$$

Each row represents the structure over an x_3 bar:

- let α_{jh_j} be a nonzero entry of the j-th row; then over $A_j^{(3)}$ (the corresponding 3-bar) of B there lie exactly α_{jh_j} 2-bars;
- the value of each $a_{i,j}$ is the number of x_1-bars over the j-th 2-bar of the j-th 3-bar.

Moreover, we set the following two conditions, holding on the entries of the matrix for $i = 1, ..., k-1$ $j = 1, ..., \alpha_1 - 1$:

1. $a_{i,j} > a_{i,j+1}$;
2. $a_{i,j} \geq a_{i+1,j}$;
3. $\sum_i^{k-1}\sum_j^{\alpha_1-1} a_{i,j} = p$.

The number of nonzero entries is clearly h. From now on we use these matrices, that we call *IP-type* associated to (p,h,k)[7], in order to count strongly stable monomial ideals, with $H\_(t) = p$.

[7]This name stands for *integer partition type*, since we will connect them to the classical theory of integer partitions.

Proposition 5.9.18. There is a one-to-one correspondence between strongly stable Bar Codes in three variables, with bar list (p, h, k) and IP-type matrices associated to (p, h, k).

Proof: Consider a strongly stable Bar Code B, with bar list (p, h, k), associated to $\mathsf{N}(J)$, the Groebner escalier of the strongly stable ideal J.
Its h x_2-bars are distributed as $(l_2(A_1^{(3)}), ..., l_2(A_k^{(3)}))$ and, by proposition 5.9.3, $l_2(A_1^{(3)}) > ... > l_2(A_k^{(3)})$.
We associate to B a $(k \times l_2(A_1^{(3)}))$-matrix with the same procedure as above. More precisely:

$$a_{i,j} = \begin{cases} 0 \text{ if } j < i \\ l_1(B_{i,j}^{(2)}) \text{ otherwise, where } B_{i,j}^{(2)} \text{ is the } i\text{-th 2-bar over the } j\text{-th 3-bar.} \end{cases}$$

The relation $\sum_i^{k-1} \sum_j^{\alpha_1 - 1} a_{i,j} = p$ is a straightforward consequence of the definition of Bar Code.
Since $l_2(A_1^{(3)}) > ... > l_2(A_k^{(3)})$, each row is shifted to the right of one entry and, again by proposition 5.9.3, $a_{i,j} > a_{i,j+1}$.
We only have to prove that $a_{i,j} \geq a_{i+1,j}$.
By the previous comments, the case $a_{i,j} = 0$, $a_{i+1,j} \neq 0$ cannot occur.
If $a_{i,j} \neq 0$, $a_{i+1,j} = 0$ there is nothing to prove, so we only have to deal with the case $a_{i,j}, a_{i+1,j} \neq 0$.
The value $a_{i,j}$ means that $x_1^{a_{i,j}-1} x_2^{\beta} x_3^{i-1} \in \mathsf{N}(J)$, lying over $A_i^{(3)}$ is the $(\beta+1)$-th term of the 2-bars lying over $A_i^{(3)}$ and also that $x_1^{a_{i,j}} x_2^{\beta} x_3^{i-1} \notin \mathsf{N}(J)$. Similar comments hold for $a_{i+1,j}$, for which $x_1^{a_{i+1,j}-1} x_2^{\beta-1} x_3^{i} \in \mathsf{N}(J)$. Suppose by contradiction $a_{i,j} < a_{i+1,j}$.
By the strongly stable property, $x_1^{a_{i+1,j}-1} x_2^{\beta} x_3^{i-1} \in \mathsf{N}(J)$, but this is impossible since

$$x_1^{a_{i+1,j}-1} x_2^{\beta} x_3^{i-1} \mid x_1^{a_{i,j}} x_2^{\beta} x_3^{i-1} \notin \mathsf{N}(J).$$

Let now M be an IP-type $(k \times \alpha_k)$-matrix, with h nonzero entries and $\sum_i^{k-1} \sum_j^{\alpha_1 - 1} a_{i,j} = p$. We associate M to a Bar Code B as follows:

- we draw k 3-bars, one for each row of the matrix;
- we draw over the i-th 3-bar as many 2-bars as the number of nonzero entries in the i-th row of M;
- we conclude drawing over the j-th 2-bar as many 1-bars as the value of the nonzero entry corresponding to the j-th x_2-bar in the matrix.

By construction, the above B is admissible.
Moreover, since $\sum_i^{k-1} \sum_j^{\alpha_1 - 1} a_{i,j} = p$, we have exactly p 1-bars, so we are representing the

Groebner escalier $\mathsf{N}(J)$ of a zerodimensional ideal such that $H\_(t, J) = p$.
We prove that it is strongly stable. More precisely, for each $x_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3} \in \mathsf{N}(J)$, we need to prove that

1. $x_1^{\alpha_1}x_2^{\alpha_2+1}x_3^{\alpha_3-1} \in \mathsf{N}(J)$;
2. $x_1^{\alpha_1+1}x_2^{\alpha_2}x_3^{\alpha_3-1} \in \mathsf{N}(J)$;
3. $x_1^{\alpha_1+1}x_2^{\alpha_2-1}x_3^{\alpha_3} \in \mathsf{N}(J)$.

Point 1 is clearly true by $a_{i,j} \geq a_{i+1,j}$. Indeed $\alpha_3 - 1$ and α_3 represent two consecutive rows and $\alpha_2 + 1$ and α_2 represent the same column by the shifting. We are requiring that there is α_1 in the position identified by $\alpha_3 - 1, \alpha_2 + 1$.
Similarly, point 2 is true by $a_{i,j} > a_{i,j+1}$ and $a_{i,j} \geq a_{i+1,j}$, whereas $a_{i,j} > a_{i,j+1}$ ensures point 3. $\diamond$

Thanks to this proposition, we can find a bound for the number of strongly stable Bar Codes with $H\_(t) = p$.
For this purpose, we need some definitions from the theory of plane partitions.

Definition 5.9.19. A *plane partition* α of a positive integer $p \in \mathbb{N}$, is a partition of p in which the parts have been arranged in a 2-dimensional array.
Such an array is weakly decreasing across rows and down columns. Different configurations are regarded as different plane partitions.
A plane partition α is called *row strict* if it is decreasing on the rows and *column strict* if it is decreasing on the columns.
We call *shape* of the plane partition α the list $(\alpha_1, ..., \alpha_k)$, where α_i is the number of entries for the i-th row of the array, $i = 1, ..., k$.

Conventionally, the zero values in the table are not written down and they are replaced by blanks.

Example 5.9.20. The matrix

$$\begin{matrix} 4 & 2 & 1 \\ 3 & 1 \end{matrix}$$

represents a plane partition of $p = 11$ with shape $(3, 2)$. Such a plane partition is simultaneously row strict and column strict.
The following plane partition of $p = 12$ is only column strict

$$\begin{matrix} 4 & 4 \\ 3 & 1 \end{matrix}$$

and its shape is $(2, 2)$.

Definition 5.9.21. A *strict shifted plane partition* is a plane partition such that each row is indented only one space w.r.t. the previous row and

- rows are weakly decreasing (from the left to the right);
- columns are strictly decreasing (from the top to the bottom).

Example 5.9.22. The plane partition

$$\begin{array}{ccc} 4 & 4 & 3 \\ & 3 & 1 \end{array}$$

is a strict shifted plane partition of $p = 15$.

Definitions 5.9.19 5.9.21 and are classical definitions, found in literature.
For our purpose, we require our partitions to be:

- such that the rows will be indented potentially more than one space;
- weakly decreasing down columns;
- strictly decreasing across rows.

Definition 5.9.23. The *hook length* of an entry c in a matrix M is the following sum:

$$h(c) = d(c) + s(c) + 1$$

where $d(c)$ are the entries on the right of c, while $s(c)$ is the number of entries below c.

We also need the following

Lemma 5.9.24. If a plane partition is an array as

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3}\dots \\ \vdots & \vdots & \vdots\dots \\ a_{1,2} & a_{2,2} & 0 \\ a_{1,3} & 0 & 0 \end{pmatrix}.$$

and it is column strict, then it contains the arrays of the form

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3}\dots. \\ \vdots & \vdots & \vdots\dots \\ 0\dots & a_{2,2} & a_{2,3} \\ 0 & 0\dots & a_{3,3} \end{pmatrix}.$$

Proof: With the previous notation, since $a_{i,j} > a_{i,j+1} \geq a_{i+1,j+1}$, then $a_{i,j} > a_{i+1,j+1}$, for $i = 1, ..., k-1$ $j = 1, ..., \alpha_1 - 1$. $\diamond$

Remark 5.9.25. The correspondence between the two plane partitions of lemma 5.9.24 is not a bijection since for example

$$\begin{pmatrix} 12 & 8 & 6 \\ 11 & 5 & 0 \end{pmatrix}.$$

cannot be shifted to the right.

If λ is the integer partition of $\mu(2)$, giving the shape of the matrix, we can give the generating function of column strict plane partitions of p with shape λ, namely

$$q^{N(\lambda)} \prod_c \frac{1}{1-q^{h(c)}},$$

where c is an entry of the matrix, $h(c)$ its hook length and $N(\lambda) = \sum_i i\lambda_i$.
This function gives the number of matrices of shape λ arranged by weight: its Taylor series at a given degree p gives the number of matrices of weight p.
Let us see a trivial example.

Example 5.9.26. Let $p = 4$; we want to count the number of matrices of type

$$\begin{pmatrix} A & B \\ C & 0 \end{pmatrix}$$

with $A + B + C = 4$.
It turns out that there exists only one matrix of this kind, namely

$$\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}.$$

We now count the number of matrices of the above type via the generating function. The shape of the matrix is $(2, 1)$.
We have $h(A) = 3$, $h(B) = h(C) = 1$ and the the generating function turns out to be

$$q^4 \frac{1}{1-q^3} \frac{1}{(1-q)^2}.$$

If we take its Taylor series at degree 4 we have exactly 1.
In this case, the number of these plane partitions coincides with the one of the particular plane partitions we are looking for, even if it is not true in the general case, where we only obtain an upper bound.

Now we apply these facts to a very precise affine Hilbert polynomial, making detailed computations.

Example 5.9.27. Let us count the number of strongly stable ideals in $\mathbf{k}[x_1, x_2, x_3]$ having affine Hilbert polynomial $H\_(t) = 10$.
First of all, we enumerate the bar-lists. There are bar lists of the form $(10, h, 1)$, for $h = 1, ..., 4$. Then, there are others of the form $(10, h, k)$ where $k = 2, 3$. Indeed, we cannot find a partition of $10 = \frac{4\cdot 5}{2}$ in 4 parts, such that their minimal sum is smaller or equal than 10, whereas for $k = 3$ we can find a partition of $6 = \frac{3\cdot 4}{2}$ in 3 parts with minimal sum smaller or equal than 10, namely $6 = 3 + 2 + 1$, $Sm([3, 2, 1]) = 10$. For $k = 2$ we have $\min(\alpha) = 3 = \frac{2\cdot 3}{2}$ and $\max(\alpha) = 5$ since there are no partitions of 6 into two distinct parts with minimal sum smaller or equal than 10, whereas there is one for 5, i.e. $5 = 3 + 2$, $Sm([3, 2]) = 9$.
We repeat for $k = 3$ finding $\min(\alpha) = 6 = \max(\alpha)$. The bar-lists are then:

1. $(10, 1, 1)$;
2. $(10, 2, 1)$;
3. $(10, 3, 1)$;
4. $(10, 4, 1)$;
5. $(10, 3, 2)$;
6. $(10, 4, 2)$;
7. $(10, 5, 2)$;
8. $(10, 6, 3)$;

For $1, 2, 3, 4$ above, i.e.

1. $(10, 1, 1)$;
2. $(10, 2, 1)$;
3. $(10, 3, 1)$;
4. $(10, 4, 1)$.

We proceed as in 2 variables: $Q(10, 1) + Q(10, 2) + Q(10, 3) + Q(10, 4) = 10$.
Consider now $(10, 3, 2)$.
Since $3 = 2 + 1$ we only have matrices of type

$$M = \begin{pmatrix} a_{1,1} & a_{1,2} \\ 0 & a_{2,2} \end{pmatrix}.$$

We shift and we get the hook lengths

$$M = \begin{pmatrix} 3 & 1 \\ 1 & 0 \end{pmatrix},$$

while $N(\lambda) = 4$, so the generating function we have to examine is $q^4 \frac{1}{1-x^3} \frac{1}{(1-x)^2}$ and we get a bound of 12 strongly stable Bar Codes.
Direct computation shows that their actual number in this case is 7.
Take then $(10, 4, 2)$
Since $4 = 3 + 1$, we only have to deal with these matrices

$$M = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} & 0 \end{pmatrix},$$

leading to the following hook numbers

$$M = \begin{pmatrix} 4 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix},$$

$N(\lambda) = 5$ and the generating function $q^5 \frac{1}{(1-q^4)} \frac{1}{(1-q^2)} \frac{1}{(1-q)^2}$. In conclusion we get a bound of 14 over 5 real strongly stable Bar Codes.
Consider now $(10, 5, 2)$. We have $5 = 4 + 1 = 3 + 2$, so we would have two cases to examine but, since $Sm([4, 1]) > 10$, we only deal with the second partition, getting the matrices

$$M = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} & a_{2,3} \end{pmatrix}.$$

and

$$M = \begin{pmatrix} 4 & 3 & 1 \\ 2 & 1 & 0 \end{pmatrix}.$$

Since $N(\lambda) = 7$, we have $q^7 \frac{1}{(1-q^4)(1-q^3)(1-q^2)(1-q)^2}$, from which we get a bound of 7 strongly stable Bar Codes. Their actual number is 1 (again by direct computation).
We conclude with $(10, 6, 3)$, for which by $6 = 3 + 2 + 1$. We obtain the matrices

$$M = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} & a_{2,3} \\ 0 & 0 & a_{3,3} \end{pmatrix}.$$

and

$$M = \begin{pmatrix} 5 & 3 & 1 \\ 3 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Since $N(\lambda) = 10$ we have $q^{10}\frac{1}{(1-q^5)(1-q^3)^2(1-q)^3}$ leading to 1 which is simultaneously the bound and the exact number.
In conclusion we have exactly 24 strongly stable ideals in 3 variables with constant affine Hilbert polynomial $H\_(t) = 10$ and our bound returns 44.

This work is still in progress. As shown by example 5.9.27, it would be good to sharpen this bound and I think it could be done by concentrating our study on generating functions for plane partitions. Moreover, we are studying a generalization to n variables.

CHAPTER 6

J-marked bases and J-marked families.

6.1 Introduction.

In this chapter, ideals $I \triangleleft \mathcal{P}$ are examined from another point of view.
Indeed, while our previous studies were mainly focused on the Groebner escalier $\mathsf{N}(I)$, now our starting point is a generating set for I, which in general is not the monomial basis $\mathsf{G}(I)$ (so multiple terms are allowed).
In particular, we deal with the following problem

Problem 6.1.1. *Given any monomial ideal $J \triangleleft \mathcal{P}$ find a characterization for the family $\mathcal{M}f(J)$ of all homogeneous ideals $I \triangleleft \mathcal{P}$ such that the basis of $\mathcal{P}/I$ is given by the set of terms in the Groebner escalier $\mathsf{N}(J)$ of J.*

The most relevant examples of ideals $I \in \mathcal{M}f(J)$ are the ideals I such that $In_<(J)$ w.r.t. some term-ordering $<$, but in general these form a proper subset of $\mathcal{M}f(J)$. Therefore, we

must overcome the Groebner framework.
A computational description of the whole family $\mathcal{M}f(J)$ is obtained in [8, 27] for J strongly stable. These families are optimal for many applications, for instance for an effective study of Hilbert schemes (see [10]).
In section 6.2, we recall the main results of [8, 27] and we explain how the connected algorithms, described and analyzed in the above two papers can be concretely implemented in Singular [30]. This is a work done in collaboration with F. Cioffi, W. Decker, H. Schoenemann and M. Roggero.
Then, we relax both the assumption of polynomial ring over a field allowing polynomial rings over any commutative ring and the assumption of strong stability for J, allowing any monomial ideal, so we pass from $\mathcal{P} = \mathbf{k}[x_1, ..., x_n]$ to $\mathcal{Q} := A[x_1, ..., x_n]$, where A is any commutative ring. We address then the problem below:

Problem 6.1.2. *Given any monomial ideal $J \lhd \mathcal{Q}$, find a characterization for the family $\mathcal{M}f(J)$ of all homogeneous ideals $I \lhd \mathcal{Q}$ such that the A-module $\mathcal{Q}/I$ is free with basis given by the set of terms in the Groebner escalier $\mathsf{N}(J)$ of J.*

In this chapter, we give then an overall view on what can be said about the above question for an *arbitrary* monomial ideal J, enhancing some ideas introduced by Janet in [54, 55, 56].
This is a joint work with Teo Mora and Margherita Roggero [19].
The main ideas we deal with are those of *multiplicative variable* and *complete system*, leading to the so called *Janet decomposition* for terms (see section 6.3). These concepts date back to the late nineteenth century and the first decades of the twentieth one. In the historical note at section 6.7 we present a detailed overview of their appearances, evolution and applications.
By exploiting the he Bar Code B_M associated to a finite set of terms $M = \{\tau_1, ..., \tau_m\} \subset \mathcal{Q}$, it is very simple to find the multiplicative variables of each τ_j, $j = 1, ..., m$ and to detect the completeness of M.
Both the multiplicative variables of $\tau_j \in M$ and the completeness of M itself strongly depend on the order given to the variables.
A problem one can face is:

Problem 6.1.3. Given a finite set of terms $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$ is there any *ordering on the variables* $x_1, ..., x_n$ such that M is *complete*?

We will show that also this problem can be solved exploiting the Bar Code (6.3).
In Janet's theory the ideals I are generated by the so called *involutive bases* (after Zarkov). Indeed, Janet develops his ideas assuming to be in *generic coordinates*. Hence the homogeneous ideals I and J he considers satisfy many good properties that always hold after

having performed a *generic linear change of coordinates*. In particular, J is the generic initial ideal of I w.r.t. the (deg)-revlex ordering.

From a computational point of view, a general change of coordinates is remarkably heavy. For this reason, we are interested to enhance the theory, getting rid of the generic coordinate assumption. Indeed, Janet's ideas permit to go beyond this context and to recover results and techniques of both Groebner basis theory and J-marked basis theory. In fact we do not need to impose a term-ordering on the given polynomial ring.

We identify two essential features that are key points for most computations in both the above frameworks:

I) I is generated by a set of polynomials, marked on the terms of a suitable generating set of the monomial ideal J;

II) there is a reduction process w.r.t. these marked polynomials, that is used to rewrite each element of P/I as an element of the free A-module $\langle \mathsf{N}(J) \rangle$.

Janet's notions of multiplicative variable and complete system allow to construct such marked set of generators for I and to define an efficient reduction process.
We examine and compare two different definitions of multiplicative variable given by Janet in [54, 55] and in [56], that are equivalent in general coordinates. We underline similarities and differences and introduce the notion of *stably complete* set of terms, when both conditions hold. We show that every monomial ideal J has only one stably complete set of generators (possibly made of infinitely many terms) that we called *star set* and denoted by $\mathcal{F}(J)$ (5.3, 6.4).

Furthermore, we define a reduction procedure with respect to a homogeneous set of polynomials marked on a stably complete system $\mathcal{F}(J)$ and prove its noetherianity (6.5). As a consequence we are able to give a first, general answer to Problem 6.1.2 .

Of course, the most interesting cases are those of ideals J such that their generating stably complete set M is finite. We prove that they are the *quasi stable* ideals (6.4) and that $\mathcal{F}(J)$ is their Pommaret basis. Among them, those such that $\mathcal{F}(J)$ coincides with the monomial basis are exactly the *stable* ones.
For the class of quasi stable ideals J we give a more complete and effective answer to Problem 6.1.2. Indeed, we prove that our description of $\mathcal{M}f(J)$ is natural, in the sense that it defines a representable functor from the category of $\mathbb{Z}$-algebras to the category of sets. We give then an effective procedure computing equations for the scheme that represents this functor (c.f. 6.6).
Moreover, switching to our usual point of view on ideals, so mainly dealing with the Groebner escalier, we show how to generalize Moeller algorithm in order to obtain an involutive

basis for a zerodimensional radical ideal, starting with the associated finite set of distinct points (see 6.6).

6.2 Singular libraries on strongly stable ideals and marked bases.

In the papers [8, 27], given a strongly stable monomial ideal J, the authors study the families

$$\mathcal{M}f(J) := \{I \triangleleft \mathcal{S},\ \text{such that } \mathcal{S} = I \oplus \mathsf{N}(J) \text{ as } k - \text{vector space}\},$$

and they establish what are the conditions making $\mathcal{M}f(J)$ an affine scheme.
In order to study these families, they introduce some special homogeneous polynomials, called J-marked polynomials, naming J-marked sets the sets of J-marked polynomials.
A J-marked set $\mathcal{G}$ such that $I := (\mathcal{G}) \in \mathcal{M}f(J)$ is called J-marked basis.
If J is a strongly stable monomial ideal[1] such a basis shares many properties with the homogeneous reduced Groebner basis .
Then, they define a reduction procedure and a Buchberger-like criterion, in order to decide whether a J-marked set is a J-marked basis or not.
Moreover, they prove that there is a biunivocal correspondence between the ideals $I \in \mathcal{M}f(J)$ and the points of an affine scheme, consequentely named J-marked scheme in [27].
Basing on the theory developed in these papers, we implemented two libraries in order to study J-marked bases and J-marked schemes.
They have been written in the programming language provided by the open source computer algebra system Singular ([30]) and integrated in the 3-1-6 release of this software.
More precisely:

- `JMBTest.lib` ([17]) is a library which checks whether a J-marked set $\mathcal{G}$ is a J-marked basis or not;
- `JMSConst.lib` ([18]) is a library which computes the equations of the J-marked scheme associated to a strongly stable monomial ideal J.

In this section, we recall the theory underlying the implementations and we explain the libraries themselves.
In the next sections, while talking about involutiveness, we will generalize most of the notions below to monomial ideals satisfying weaker properties than the strongly stable one.

[1] It is enough for J to be strongly stable, we do not require J to be zerodimensional as was in 5.9.

The source code can be found in appendix A.
Let us start recalling the concept of J-normal form.

Definition 6.2.1. Given a monomial ideal $J \triangleleft \mathcal{S}$ and an ideal $I \triangleleft \mathcal{S}$, a *$J$-normal form* modulo I of a polynomial $h \in \mathcal{S}$ is a polynomial $h_0 \in \mathcal{S}$ such that $h - h_0 \in I$ and $Supp(h_0) \subseteq \mathsf{N}(J)$.

Clearly, if I is an homogeneous ideal, also the J-normal from modulo I of an homogeneous polynomial turns out to be homogeneous.

Definition 6.2.2. A marked polynomial is a polynomial $f \in \mathcal{S}$, with a specific term in $Supp(f)$ that we call *head term* of f, denoting it $Ht(f)$.

We denote by

$$\mathcal{G} = \{f_\tau = \tau - \sum c_{\tau\gamma} x^\gamma,\ Ht(f_\tau) = \tau\}$$

a finite set of homogeneous marked polynomials

Definition 6.2.3. The set $\mathcal{G}$ is called *J-marked set* if the head terms $Ht(f_\tau)$ constitute the monomial basis $\mathsf{G}(J)$ of a given J and all the x^γ are in $\mathsf{N}(J)$.
A J-marked set $\mathcal{G}$ is a *J-marked basis* if $\mathsf{N}(J)$ is a basis of $\mathcal{S}/\mathcal{G}$ as a k-vector space.

We usually denote by $\mathcal{G}_p$ the degree p part of $\mathcal{G}$.
Given a set $\mathcal{G}$ of J-marked polynomials, the Singular library `JMBTest.lib` ([17]) checks whether such a set is a J-marked basis or not.
The output is a boolean value: 1 for true, 0 for false, following the usual conventions.
In order to increase the computation's speed, the input marked polynomials are arranged by degree, as a list of lists of polynomials: $\mathcal{G} = [\mathcal{G}_{\mathsf{a}_J}, ..., \mathcal{G}_{\mathsf{a}_J+h}]$, where a_J is the minimal degree for a homogeneous polynomial in the given J-marked set $\mathcal{G}$ and $\mathsf{a}_J + h$ is clearly their maximal degree.
The head terms of the elements in $\mathcal{G}$ have to make up the monomial basis $\mathsf{G}(J)$ of a strongly stable ideal J and we think them ordered with respect to a degree compatible term order.
In the procedure, our usual variable ordering $x_1 < x_2 < ... < x_n$ (or $x < y < z$ in the case of three variables or less) is supposed.
Since the head terms we choose for the input $\mathcal{G}$ are *not necessarily* the leading terms of the given polynomials with respect to any term order (see [27] for more details), it is necessary to highlight them precisely and this makes essential the introduction of a *new data type*, satisfying this requirement.
In `JMBTest.lib`, a new data type for Singular, i.e. `jmp`, the J-marked polynomial, is then introduced.

Example 6.2.4. To define $r_3 = \mathbf{zy^2} - x^2y \in \mathbf{k}[x, y, z]$, $Ht(r_3) = zy^2$ we type:
`jmp r3;`
`r3.h=` $z * y^2$`;`
`r3.t=-`$x^2 * y$`;`
where the suffix `.h` identifies the head terms, while the suffix `.t` identifies the tails.

Definition 6.2.5. We call J-marked family, the family $\mathcal{M}f(J)$ containing all the homogeneous ideals I such that $\mathsf{N}(J)$ is a basis $\mathcal{S}/I$ as a k-vector space.

Given a homogeneous ideal I and fixed a term ordering $<$, if $In_<(I) = J$, then $I \in \mathcal{M}f(J)$, but in general also other ideals belong to a J-marked family.

Proposition 6.2.6 ([27])**.** If $\mathcal{G}$ is a J-marked set, TFAE:

1. $\mathcal{G}$ is a J-marked basis;
2. $(\mathcal{G}) \in \mathcal{M}f(J)$;
3. each polynomial $h \in \mathcal{S}$ has a unique J-normal form modulo $(\mathcal{G})$.

If $I \in \mathcal{M}f(J)$ then it obviously contains a J-marked set.
If $\mathcal{G} \subset \mathcal{S}$ is a J-marked basis, it is *unique* for $I := (\mathcal{G})$.
Since J-marked sets have better properties in the case J strongly stable (5.9.1), we place us in this case. The strongly stable property for J can be checked by examining only the elements of $\mathsf{G}(J)$ ([27]). Basing on this fact, we implemented the procedure BorelCheck, a subroutine for the main procedure of the library `JMSConst.lib`, which can also be used on his own. This subroutine takes $\mathsf{G}(J)$ and the base ring as input, returning 1 if J is strongly stable and 0 otherwise.
Its functioning is rather simple, since it iterates on the monomial basis and, $\forall \tau \in \mathsf{G}(J), \forall x_i \mid \tau, x_j > x_i$, it checks whether $\tau_{ij} := \frac{\tau x_j}{x_i}$ is in the ideal or not, breaking and reporting a failure when it detects a $\tau_{ij} \notin J$.
Given an invertible matrix $A = (a_{ij}) \in GL_n(k)$ and a polynomial $f \in \mathcal{S}$, we denote by $A(f)$ the standard action of $GL_n(k)$ on $\mathcal{S}$, under the substitution

$$x_i \mapsto \sum_j a_{ij} x_j$$

and, for $I \lhd \mathcal{S}$, $A(I) := \{A(f) | \, f \in I\}$.

The strongly stable property implies the *Borel-fixed* one, i.e. if $J \lhd \mathcal{S}$ is strongly stable, it is fixed under the action of the subgroup of lower triangular invertible matrices, the notions being equivalent in the case $char(k) = 0$ (c.f. [31, 79]).
As it will be useful to understand the whole chapter, we recall here the following

Definition 6.2.7 ([43])**.** A property holding for $A(I)$ for each matrix A in a Zariski open subset of $GL_n(k)$ is said to hold for *general* or *generic coordinates*.

Galligo's theorem ([38]) says that, if we are in generic coordinates and fixed a term order $<$, the initial ideal of some ideal I w.r.t $<$, is a constant Borel-fixed monomial ideal, conventionally denoted by $gin(I)$ and called the *generic initial ideal* of I.
The strongly stable property is very important, since it allows many different applications as, for example, tho study the Hilbert scheme [8, 27, 64].
In [27], given a J-marked set

$$\mathcal{G} = \{f_\tau = \tau - \sum c_{\tau\gamma} x^\gamma,\ Ht(f_\tau) = \tau \in \mathsf{G}(J)\}$$

the authors define a reduction process *à la Buchberger* w.r.t. $\mathcal{G}$, denoting it by $\xrightarrow{\mathcal{G}}$.

Definition 6.2.8. A reduction relation $\xrightarrow{\mathcal{G}}$ is *noetherian* if the length r of any sequence $h = h_0 \xrightarrow{\mathcal{G}} h_1 \xrightarrow{\mathcal{G}} \dots \xrightarrow{\mathcal{G}} h_r$ is bounded by an integer number $m = m(h)$.

The noetherianity says that if we continue rewriting terms according to $\xrightarrow{\mathcal{G}}$, we always obtain, after a finite number of reductions, a polynomial whose support is contained in $\mathsf{N}(J)$.
We will write $h \xrightarrow{\mathcal{G}}_* g$ if $h \xrightarrow{\mathcal{G}} g$ and $\mathrm{Supp}(g) \subset \mathsf{N}(J)$, so it is not possible to reduce anymore.
An important result of [11, 86] is that, if such a reduction process à la Buchberger is noetherian, then there exists an admissible term ordering $<$ such that

$$\{Ht(f),\ f \in \mathcal{G}\} = \{\mathsf{T}_<(f),\ f \in \mathcal{G}\}.$$

We remark that for J-marked sets the reduction can be non-noetherian.
We recall now some results from [27].

Proposition 6.2.9 ([27])**.** We have the following properties.

- If $\mathcal{G} = \{f_\tau = \tau - \sum c_{\tau\gamma} x^\gamma,\ Ht(f_\tau) = \tau \in \mathsf{G}(J)\}$ is a J-marked set, with J strongly stable, each polynomial in $\mathcal{P}$ has a J-normal form modulo $(\mathcal{G})$.
- Let J a strongly stable ideal and $\mathcal{G}$ a J-marked set. Then $\mathcal{G}$ is a J-marked basis if and only if $\mathsf{N}(J)$ is free in $\mathcal{P}/(\mathcal{G})$.
- For $I \triangleleft \mathcal{S}$ homogeneous, it holds

$$I \in \mathcal{M}f(J) \Leftrightarrow I \text{ has a } J\text{-marked basis}$$

Consider now a strongly stable monomial ideal J, a J-marked set $\mathcal{G}$, and the homogeneous ideal $I = (\mathcal{G})$. We outline the procedure of [8, 27] in order to determine a J-normal form of an homogeneous polynomial modulo $\mathcal{G}$.
This is the basis for our Singular libraries.
First of all, for each degree m we define

$$W_m = \{x^\delta f_\alpha,\ deg(x^\delta) + deg(Ht(f_\alpha)) = m,\ f_\alpha \in \mathcal{G}\},$$

letting $Ht(x^\delta f_\alpha) = x^\delta Ht(f_\alpha)$, W_m is a J-marked set.
Then, denoted by $\mathsf{a}_J := \min\{m \in \mathbb{N},\ I_m \neq (0)\}$, we define for $m = \mathsf{a}_J, ..., s$ (c.f. [27]):

$$\begin{cases} V_m := \mathcal{G}_m \text{ for } m = \mathsf{a}_J \\ V_m := \mathcal{G}_m \cup \{g_\beta\ :\ x^\beta \in J_m \setminus \mathcal{G}_m\} \text{ for } m > \mathsf{a}_J \end{cases}$$

where $g_\beta := x_i g_\epsilon$ with $x_i = \min(x^\beta)$ and g_ϵ the unique polynomial of V_{m-1} whose head term is exactly $x^\epsilon = x^\beta / x_i$.
The procedure of TestJMark.lib constructing the polynomials V_m is VConst, which follows the algorithm VConstructor of [27].
VConst takes $\mathcal{G}$ as input, together with an integer number c, representing the maximal degree for the V_m's we need to construct[2].
The output is a list V, containing the polynomials V_m's, arranged by degree. More precisely, since actually each $g \in V_m$ is the product of a marked polynomial $f_\tau \in \mathcal{G}$ by a term $\sigma \in \mathcal{T}$ such that $\max(\sigma) \leq \min(\tau)$, we store only:

- $\sigma \in \mathcal{T}$;
- the position of the marked polynomial f_τ in the list $\mathcal{G}$.

The polynomials in V_m are constructed iteratively on the degree, from the minimal one, a_J, to the required c.
The polynomials in V_{a_J} are exactly the ones in $\mathcal{G}_{(\mathsf{a}_J)}$. For each $j = \mathsf{a}_J + 1, ..., c$, we add to the elements of $\mathcal{G}_j$[3] all the products of polynomials $f_\tau \in V_{j-1}$'s by the variables $x_i \leq \min(\tau)$.
Each V_m can be equipped with a total ordering $\succeq_m$ according to the following rules.

1. Considered two polynomials $f_\alpha, f_{\alpha'} \in \mathcal{G}$ set

$$f_\alpha \leq_{\min} f_{\alpha'} \Leftrightarrow deg(f_\alpha) \leq deg(f_{\alpha'}) \text{ or } deg(f_\alpha) = deg(f_{\alpha'})$$

and

$$\min\Big(\frac{Ht(f_\alpha)}{GCD(Ht(f_\alpha), Ht(f_{\alpha'}))}\Big) \leq \min\Big(\frac{Ht(f_{\alpha'})}{GCD(Ht(f_\alpha), Ht(f_{\alpha'}))}\Big).$$

[2]The criterion to determine the value of c will be explained in what follows.
[3]We remark that, possibly, $\mathcal{G}_j = \emptyset$.

2. Let $x^{\delta} f_{\alpha}, x^{\delta'} f_{\alpha'} \in W_m$, then

$$x^{\delta} f_{\alpha} \succeq_m x^{\delta'} f_{\alpha'} \Leftrightarrow x^{\delta} >_{Lex} x^{\delta'} \text{ or } x^{\delta} = x^{\delta'} \text{ and } f_{\alpha} \geq f_{\alpha'}.$$

Given the list V, obtained running VConst, the subroutine OrderingV produces the ordering induced by the two rules above. It depends on GJmpMins which deals with rule 1. and TernCompare which deals with rule 2.
Since each element of V has not been encoded as a jmp, but with a term and the position of its related polynomial of $\mathcal{G}$, the procedure directly deals with this information and does not need concretely to construct the involved jmp's.
The polynomials in V are fundamental both for the J-marked basis test and for the J-marked scheme constructor because of the following

Proposition 6.2.10. Let J be a strongly stable monomial ideal, $\mathcal{G}$ a J-marked set and $I = (\mathcal{G})$. Each term $\tau \in J_m \setminus \mathcal{G}_m$ can be reduced to a J-normal form modulo $\mathcal{G}$ using V_m and the reduction procedure is noetherian in $\mathcal{S}_m$.

The first version of the Buchberger-like criterion for J-marked families is:

Theorem 6.2.11 ([27])**.** Let J a strongly stable monomial ideal $\mathcal{G}$ a J-marked set and $I = (\mathcal{G})$. Then, $\forall f_{\tau}, f_{\tau'} \in \mathcal{G}$:

$$I \in \mathcal{M}f(J) \Leftrightarrow S(f_{\tau}, f_{\tau'}) \xrightarrow{V_m}_{*} 0.$$

Such a criterion has been enhanced in [8], via the introduction of the star decomposition and Eliahou-Kervaire S-polynomials.

Definition 6.2.12 ([8, 33])**.** Given a strongly stable monomial ideal J in $\mathcal{S}$, with monomial basis $\mathsf{G}(J)$, and a monomial $x^{\gamma} \in J$, we define $x^{\gamma} = x^{\alpha} *_J x^{\eta}$, with $\gamma = \alpha + \eta$, $x^{\alpha} \in \mathsf{G}(J)$ and $\min(x^{\alpha}) \geq \max(x^{\eta})$. Such a decomposition *exists* and is *unique*.

Definition 6.2.13. Given a J-marked set $\mathcal{G}$, a couple of polynomials $f_{\alpha}, f_{\beta} \in \mathcal{G}$, with $Ht(f_{\alpha}) = x^{\alpha}$, $Ht(f_{\beta}) = x^{\beta}$, is called *Eliahou-Kervaire couple* if it holds: $x_j x^{\alpha} = x^{\beta} *_J x^{\eta}$ for some $x_j > \min(x^{\alpha})$. The S-polynomials between an Eliahou-Kervaire couple of polynomials f_{α}, f_{β} are called *Eliahou-Kervaire S-polynomials* of $\mathcal{G}$ and they are denoted by $S^{EK}(f_{\alpha}, f_{\beta})$. By definition $S^{EK}(f_{\alpha}, f_{\beta}) = x_j f_{\alpha} - x^{\eta} f_{\beta}$, for some $x_j > \min(x^{\alpha})$, with $x_j x^{\alpha} = x^{\beta} *_J x^{\eta}$.

The Eliahou-Kervaire polynomials (or EK-polynomials, see [8, 33]) are constructed by the procedures below, which arise from the star product:

- EKCouples, which checks whether a couple of terms is an Eliahou-Kervaire couple;

- EKPolys, which construct all the EK-couples, given the input J-marked set $\mathcal{G}$;
- EKPolynomials, which finally computes the Eliahou-Kervaire polynomials.

While constructing the EK-polynomials, we also keep track of their maximal degree s.

Proposition 6.2.14 ([8]). With the usual notation it holds $I \in \mathcal{M}f(J)$ if and only if for each EK-polynomial computed by $\mathcal{G}$, it holds $S^{EK}(f_\alpha, f_\beta) \xrightarrow{V_m}_* 0$.

Once we have obtained the polynomials of the list V and the EK-polynomials, the last step consists of reducing each EK-polynomial q of degree m with respect to the V_m's, via a Buchberger-type reduction denoted by $\xrightarrow{V_m}$. If one of these EK-polynomials does not reduce to 0, the algorithm breaks and reports a negative outcome.
Given then a J-marked set $\mathcal{G}$, we can then summarize the steps executed by the main function TestJMark in `JMBTest.lib` as follows:

1. if $\mathcal{G}$ contains only one polynomial return 1[4];
2. if not, perform the following steps:
   a. compute the list E of the EK-couples and keep track of their degree;
   b. store the minimal degree $\mathtt{a}_J$ of the elements of $\mathcal{G}$ (i.e. the degree of its *first* element) and store also the maximal degree s of the EK-polynomials found in the previous step;
   c. compute $V_{\mathtt{a}_J}, V_{\mathtt{a}_J+1}, ..., V_s$;
   d. for i from 1 to $|E|$, compute the i-th EK-polynomial q corresponding to the i-th EK-couple stored in E and denote by w be its degree;
   e. reduce q w.r.t V_w, returning 0 and breaking if the reduction does not produce 0 and going again to step d. otherwise;

The Buchberger-type reduction is performed via the Singular command `reduce`, in order to take advantage of its potentialities. In order to make the procedure `reduce` individuate the head terms (which can eventually not be compatible with any term order), we multiply them by a fictitious variable, much greater than x_n.
We display now two examples of execution for `JMBTest.lib`. The first is very simple and presented with some more comments. The second is heavier from a computational point of view and it is displayed with its execution time.

[4]If only one polynomial $r1$ is given in input, the function automatically gives positive answer, since a single polynomial is surely a J-marked basis. Clearly this situation happens under the hypothesis that the ideal J of $\mathcal{S}$ is a *principal* strongly stable ideal.

Example 6.2.15. Let us start with a very simple example:

```
ring r = 0, (x,y,z),rp;
jmp r1;
r1.h= z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2 * y;
r2.t=poly(0);
jmp r3;
r3.h=z * y^2 ;
r3.t=-x^2 * y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G1=list(list(r1,r2,r3),list(),list(r4));
```

Executing our test we obtain that it is not a J-marked basis:

```
TestJMark(G1,r);
```

$\Rightarrow 0$.

In fact, the three EK-polynomials are $S_1^{EK} = 0$, $S_2^{EK} = -x2yz$, $S_3^{EK} = x2y4$, while the V polynomials are:

$V_3 = \{\mathbf{y2z} - x^2y, yz^2, z^3\}$

$V_4 = \{\mathbf{xy^2z} - x^3y, xyz^2, xz^3, \mathbf{y^3z} - x^2y^2, y^2z^2, yz^3, z^4\}$

$V_5 = \{y^5, \mathbf{x^2y^2z} - x^4y, x^2yz^2, x^2z^3, \mathbf{xy^3z} - x^3y^2, xy^2z^2, xyz^3,$
$\mathbf{y^4z} - x^2y^3, y^3z^2, y^2z^3, xz^4, yz^4, z^5\}$

$V_6 = \{xy5, \mathbf{x^3y^2z} - x^5y, x^3yz^2, x^3z^3, y^6, \mathbf{x^2y^3z} - x^4y^2, x^2y^2z^2, x^2yz^3,$
$\mathbf{xy^4z} - x^3y^3, xy^3z^2, xy^2z^3, \mathbf{y^5z} - x^2y^4, y^4z^2, y^3z^3, x^2z^4, xyz^4, y^2z^4, xz^5, yz^5, z^6\}$. Since S_2^{EK} does not go to zero, $G2F$ is not a J-marked basis.

Example 6.2.16. Consider now the following polynomials, proposed in [9]

$f_1 := \mathbf{x_5^2} + 4x_1^2 + \frac{17}{3}x_1x_2 - \frac{83}{12}x_1x_3 - \frac{23}{4}\,x_2x_3,$

$f_2 := \mathbf{x_4x_5} - \frac{3}{4}\,x_2x_3 - \frac{5}{4}\,x_1x_3 + x_1x_2,$

$f_3 := \mathbf{x_4^2} - ax_4 + x_2a + \frac{25}{6}x_2x_3 + x_2^2 + \frac{71}{18}x_1x_3 - \frac{28}{9}x_1x_2 - 5x_1^2,$

$f_4 := \mathbf{x_3x_5} - \frac{3}{4}\,x_2x_3 + \frac{3}{4}x_1x_3 - x_1x_2,$

$f_5 := \mathbf{x_3x_4} - x_2x_3,$

$f_6 := \mathbf{x_3^2} - \frac{85}{24}\,x_2x_3 - \frac{317}{72}\,x_1x_3 + \frac{71}{18}x_1x_2 + 2x_1^2,$

$f_7 := \mathbf{x_2x_5} - \frac{3}{4}\,x_2x_3 - \frac{5}{4}\,x_1x_3 + x_1x_2,$

$f_8 := \mathbf{x_2x_4} - x_2x_3 - x_1x_3 + x_1x_2,$

$f_9 := \mathbf{x_1x_5} - \frac{1}{4}x_2x_3 + \frac{1}{4}x_1x_3 - x_1x_2$,
$f_{10} := \mathbf{x_1x_4} - x_1x_2$,
$f_{11} := \mathbf{x_2^2x_3} + x_1^3$,
$f_{12} := \mathbf{x_2^3} - x_2x_3a - x_3x_1a + ax_2^2 + x_2x_1a + \frac{5}{9}x_1^3$,
$f_{13} := \mathbf{x_2x_1x_3} - \frac{11}{9}x_1^3$, $f_{14} := \mathbf{x_1x_2^2} - \frac{8}{9}x_1^3$, $f_{15} := \mathbf{x_1^2x_3} + x_1^3$, $f_{16} := \mathbf{x_1^2x_2} + \frac{2}{3}x_1^3$, $f_{17} := \mathbf{x_1^4}$.
and homogenize them, obtaining the Singular code:

```
ring r=(0,a),(x(0..5)),rp;
jmp f1;
f1.h=x(5)^2;
f1.t=4*x(1)^2+(17/3)*x(1)*x(2)-(83/12)*x(1)*x(3)-(23/4)*x(2)*x(3);
jmp f2;
f2.h=x(4)*x(5);
f2.t=-(3/4)*x(2)*x(3)-(5/4)*x(1)*x(3)+x(1)*x(2);
jmp f3;
f3.h=x(4)^2;
f3.t=-a*x(0)*x(4)+a*x(0)*x(2)+(25/6)*x(2)*x(3)+x(2)^2+(71/18)*x(1)*x(3)-
(28/9)*x(1)*x(2)-5*x(1)^2;
jmp f4;
f4.h=x(3)*x(5);
f4.t=-(3/4)*x(2)*x(3)+(3/4)*x(1)*x(3)-x(1)*x(2);
jmp f5;
f5.h=x(3)*x(4);
f5.t=-x(2)*x(3);
jmp f6;
f6.h=x(3)^2;
f6.t=-(85/24)*x(2)*x(3)-(317/72)*x(1)*x(3)+(71/18)*x(1)*x(2)+2*x(1)^2;
jmp f7;
f7.h=x(2)*x(5);
f7.t=-(3/4)*x(2)*x(3)-(5/4)*x(1)*x(3)+x(1)*x(2);
jmp f8;
f8.h=x(2)*x(4);
f8.t=-x(2)*x(3)-x(1)*x(3)+x(1)*x(2);
jmp f9;
f9.h=x(1)*x(5);
f9.t=-(1/4)*x(2)*x(3)+(1/4)*x(1)*x(3)-x(1)*x(2);
jmp f10;
```

```
f10.h=x(1)*x(4);
f10.t=-x(1)*x(2);
jmp f11;
f11.h=x(2)^2*x(3);
f11.t=x(1)^3;
jmp f12;
f12.h=x(2)^3;
f12.t=-a*x(0)*x(2)*x(3)-a*x(0)*x(3)*x(1)+a*x(0)*x(2)^2+a*x(0)*x(2)*
x(1)+(5/9)*x(1)^3;
jmp f13;
f13.h=x(2)*x(1)*x(3);
f13.t=-(11/9)*x(1)^3;
jmp f14;
f14.h=x(1)*x(2)^2;
f14.t=-(8/9)*x(1)^3;
jmp f15;
f15.h=x(1)^2*x(3);
f15.t=x(1)^3;
jmp f16;
f16.h=x(1)^2*x(2);
f16.t=(2/3)*x(1)^3;
jmp f17;
f17.h=x(1)^4;
f17.t=poly(0);
list G1V= list( list(f6,f10,f8,f5,f3,f9,f7,f4,f2,f1),
list(f16,f14,f12,f15,f13,f11), list(f17));
TestJMark(G1V,r);
```

Running TestJMark on them we obtain that this set is a J-marked basis, for all values of the parameter and the result is achieved in $4870ms$.

As it can be seen by the example 6.2.16 above, the library `JMBTest.lib` clearly works if the coefficients are numerical but also if the coefficients contain some *parameters*, provided that they are correctly defined in the ring declaration, according to Singular's syntax.

In [27] is provided the construction of an affine scheme, whose points are in *biunivocal correspondence* with the ideals $I \in \mathcal{M}f(J)$ for J strongly stable.
Taken an $x^\alpha \in \mathsf{G}(J)$, construct the polynomials $F_\alpha := x^\alpha - \sum c_{\alpha\gamma} x^\gamma$, where $x^\gamma \in N(J)_{|\alpha|}$

and the $c_{\alpha\gamma}$'s are parameters, calling $\mathbf{C}$ the set containing them and defining $N := |\mathbf{C}|$. Let $\mathcal{G}$ be the set of all the F_α, which turns out to be a J-marked set with $Ht(F_\alpha) = x^\alpha$. Using the J-marked set $\mathcal{G}$, via an unique specialization of the elements of $\mathbf{C}$ in $\mathbf{k}^N$, we can obtain the J-marked basis of every ideal $I \in \mathcal{M}f(J)$ by the uniqueness of the J-marked basis. Remember that *not all the specializations* produce an ideal of $\mathcal{M}f(J)$.
Once we have computed the analogous of the V_m polynomials, whose coefficients are allowed to be parameters, that we call $\mathcal{V}_m$, we produce the EK-couples and the analogous of the EK-polynomials for this case.
For each EK-polynomial q, $deg(q) = m$, we reduce it w.r.t. $\mathcal{V}_m$ and we consider the coefficients of the obtained polynomial as generators of an ideal $\mathfrak{I}$ of $\mathbf{k}[\mathbf{C}]$.

Theorem 6.2.17. There is a one to one correspondence between the ideals of $\mathcal{M}f(J)$ and the points of the affine scheme in $\mathbf{k}^N$ defined by the ideal $\mathfrak{I}$.

Definition 6.2.18. The affine scheme defined by the ideal $\mathfrak{I}$ is called *J-marked scheme* and it is denoted by $S(J)$.

Given the monomial basis $\mathsf{G}(J)$ of a strongly stable ideal J, arranged in a list increasingly ordered by degree, the Singular library `JMSConst.lib` computes the equations of the associated J-marked scheme (6.2.18).
It is strongly related with `JMBTest.lib`, since the criterion used in order to perform the J-marked basis test is exploited also here (c.f. 6.2.14).
Employing the calculation of the Groebner escalier, degree by degree[5], the software (more precisely the subroutine NumNewVar) computes the cardinality N of the set $\mathbf{C}$, containing the parameters and then it generates a tail for each head $\tau \in \mathsf{G}(J)$ (see the procedure NewTails).
Then, ArrangeTails reorders the obtained jmp's by degree in a list of lists $\mathcal{G}$.
Next step is, exactly as before, the computation of the EK-polynomials and of the $\mathcal{V}$ polynomials of the same required degrees.
After that, a Buchberger-type reduction is again performed on the EK-polynomials, w.r.t. the $\mathcal{V}$ polynomials of the same degree[6] and the nonzero coefficient of the resulting polynomials are precisely the equations of the required J-marked scheme, so they are collected and returned as final output.
We summarize here the steps of the main function JMarkedScheme on an input ideal J:

a. perform BorelCheck and exit if J is not strongly stable;

[5]The computation refers to the generators of J.
[6]Remember that we are dealing with homogeneous polynomials.

b otherwise, continue as follows:

1. for each generator x^α of J find the Groebner escalier of degree $|\alpha|$, $N(J)_{|\alpha|}$ and store both $N(J)_{|\alpha|}$ and its cardinality;
2. produce a J-marked set, attaching to each x^α a tail, which is a linear combination of parameters in $\mathbf{C}$, with coefficients in $N(J)_{|\alpha|}$ for each $|\alpha|$;
3. compute the list E of EK-couples, taking track of the degree of the corresponding EK-polynomials;
4. for i from 1 to $|E|$, compute the i-th EK-polynomial q corresponding to the i-th EK-couple previously stored in$E$ and let w be its degree;
5. reduce q w.r.t V_w, and store the coefficients of the reduced polynomial in a list S;
6. repeat step 5. for all the EK-polynomials.
   At the end, S contains the equations of the required scheme (see [27]).

Now we display two examples of execution of `JMSConst.lib`. As for `JMBTest.lib`, the first example is simple and provided with some comments, while the second is heavier and displayed with the execution time.

Example 6.2.19. Let us first take the simple example given by the strongly stable ideal $J = (x_1^8, x_2^2, x_1x_2, x_3)$ of $\mathbf{k}[x_0, ..., x_3]$.
The corresponding Singular code is

```
ring r=0,(x(0..3)),rp;
ideal Borid=x(3),x(1)*x(2),x(2)^2,x(1)^8;
JMarkedScheme(Borid,r);
```

[1] :
$(-c(1) * c(7) + c(1) * c(4) * c(6) - c(1) * c(4)^2 * c(5))$
[2] :
$(c(1) * c(9) + c(1) * c(5)^2)$
[3] :
$(c(1) * c(10) + c(1) * c(4) * c(9) - c(1) * c(5) * c(8) + 2 * c(1) * c(5) * c(6) - c(1) * c(4) * c(5)^2)$
[4] :
$(c(1) * c(11) + c(1) * c(4) * c(10) - c(1) * c(6) * c(8) + c(1) * c(5) * c(7) + c(1) * c(6)^2 - c(1) * c(4) * c(5) * c(6))$
[5] :
$(c(1) * c(4) * c(11) - c(1) * c(7) * c(8) + c(1) * c(6) * c(7) - c(1) * c(4) * c(5) * c(7))$
[6] :

$(c(7) - c(4) * c(6) + c(4)^2 * c(5))$

[7] :

$(-c(9) - c(5)^2)$

[8] :

$(-c(10) - c(4) * c(9) + c(5) * c(8) - 2 * c(5) * c(6) + c(4) * c(5)^2)$

[9] :

$(-c(11) - c(4) * c(10) + c(6) * c(8) - c(5) * c(7) - c(6)^2 + c(4) * c(5) * c(6))$

[10] :

$(-c(4) * c(11) + c(7) * c(8) - c(6) * c(7) + c(4) * c(5) * c(7))$

[11] :

$(-c(1) * c(20) + c(1) * c(4) * c(19) - c(1) * c(4)^2 * c(18) + c(1) * c(4)^3 * c(17) - c(1) * c(4)^4 * c(16) + c(1) * c(4)^5 * c(15) - c(1) * c(4)^6 * c(14) + c(1) * c(4)^7 * c(13) + c(1) * c(8) * c(12) - c(1) * c(6) * c(12) + 2 * c(1) * c(4) * c(5) * c(12) - c(1) * c(4)^8)$

[12] :

$(c(1) * c(7) - c(1) * c(4) * c(6) + c(1) * c(4)^2 * c(5))$

[13] :

$(c(1) * c(7) * c(13) - c(1) * c(4) * c(6) * c(13) + c(1) * c(4)^2 * c(5) * c(13) - c(1) * c(4) * c(7) + c(1) * c(4)^2 * c(6) - c(1) * c(4)^3 * c(5))$

[14] :

$(c(1) * c(7) * c(14) - c(1) * c(4) * c(6) * c(14) + c(1) * c(4)^2 * c(5) * c(14) - c(1) * c(4) * c(7) * c(13) + c(1) * c(4)^2 * c(6) * c(13) - c(1) * c(4)^3 * c(5) * c(13) + c(1) * c(4)^2 * c(7) - c(1) * c(4)^3 * c(6) + c(1) * c(4)^4 * c(5))$

[15] :

$(c(1) * c(7) * c(15) - c(1) * c(4) * c(6) * c(15) + c(1) * c(4)^2 * c(5) * c(15) - c(1) * c(4) * c(7) * c(14) + c(1) * c(4)^2 * c(6) * c(14) - c(1) * c(4)^3 * c(5) * c(14) + c(1) * c(4)^2 * c(7) * c(13) - c(1) * c(4)^3 * c(6) * c(13) + c(1) * c(4)^4 * c(5) * c(13) - c(1) * c(4)^3 * c(7) + c(1) * c(4)^4 * c(6) - c(1) * c(4)^5 * c(5))$

[16] :

$(c(1) * c(7) * c(16) - c(1) * c(4) * c(6) * c(16) + c(1) * c(4)^2 * c(5) * c(16) - c(1) * c(4) * c(7) * c(15) + c(1) * c(4)^2 * c(6) * c(15) - c(1) * c(4)^3 * c(5) * c(15) + c(1) * c(4)^2 * c(7) * c(14) - c(1) * c(4)^3 * c(6) * c(14) + c(1) * c(4)^4 * c(5) * c(14) - c(1) * c(4)^3 * c(7) * c(13) + c(1) * c(4)^4 * c(6) * c(13) - c(1) * c(4)^5 * c(5) * c(13) + c(1) * c(4)^4 * c(7) - c(1) * c(4)^5 * c(6) + c(1) * c(4)^6 * c(5))$

[17] :

$(c(1) * c(7) * c(17) - c(1) * c(4) * c(6) * c(17) + c(1) * c(4)^2 * c(5) * c(17) - c(1) * c(4) * c(7) * c(16) + c(1) * c(4)^2 * c(6) * c(16) - c(1) * c(4)^3 * c(5) * c(16) + c(1) * c(4)^2 * c(7) * c(15) - c(1) * c(4)^3 * c(6) * c(15) + c(1) * c(4)^4 * c(5) * c(15) - c(1) * c(4)^3 * c(7) * c(14) + c(1) * c(4)^4 * c(6) * c(14) - c(1) * c(4)^5 * c(5) * c(14) + c(1) * c(4)^4 * c(7) * c(13) - c(1) * c(4)^5 * c(6) * c(13) + c(1) * c(4)^6 * c(5) * c(13) +$

$c(1)*c(9)*c(12)+c(1)*c(5)^2*c(12)-c(1)*c(4)^5*c(7)+c(1)*c(4)^6*c(6)-c(1)*c(4)^7*c(5))$
[18] :

$(-c(1)*c(5)*c(20)+c(1)*c(4)*c(5)*c(19)+c(1)*c(7)*c(18)-c(1)*c(4)*c(6)*c(18)-c(1)*c(4)*c(7)*c(17)+c(1)*c(4)^2*c(6)*c(17)+c(1)*c(4)^2*c(7)*c(16)-c(1)*c(4)^3*c(6)*c(16)-c(1)*c(4)^3*c(7)*c(15)+c(1)*c(4)^4*c(6)*c(15)+c(1)*c(4)^4*c(7)*c(14)-c(1)*c(4)^5*c(6)*c(14)-c(1)*c(4)^5*c(7)*c(13)+c(1)*c(4)^6*c(6)*c(13)+c(1)*c(10)*c(12)+c(1)*c(5)*c(6)*c(12)+c(1)*c(4)^6*c(7)-c(1)*c(4)^7*c(6))$
[19] :

$(-c(1)*c(6)*c(20)+c(1)*c(4)*c(5)*c(20)+c(1)*c(7)*c(19)-c(1)*c(4)*c(7)*c(18)+c(1)*c(4)^2*c(7)*c(17)-c(1)*c(4)^3*c(7)*c(16)+c(1)*c(4)^4*c(7)*c(15)-c(1)*c(4)^5*c(7)*c(14)+c(1)*c(4)^6*c(7)*c(13)+c(1)*c(11)*c(12)+c(1)*c(5)*c(7)*c(12)-c(1)*c(4)^7*c(7))$
[20] :

$(c(20)-c(4)*c(19)+c(4)^2*c(18)-c(4)^3*c(17)+c(4)^4*c(16)-c(4)^5*c(15)+c(4)^6*c(14)-c(4)^7*c(13)-c(8)*c(12)+c(6)*c(12)-2*c(4)*c(5)*c(12)+c(4)^8)$
[21] :

$(-c(7)+c(4)*c(6)-c(4)^2*c(5))$
[22] :

$(-c(7)*c(13)+c(4)*c(6)*c(13)-c(4)^2*c(5)*c(13)+c(4)*c(7)-c(4)^2*c(6)+c(4)^3*c(5))$
[23] :

$(-c(7)*c(14)+c(4)*c(6)*c(14)-c(4)^2*c(5)*c(14)+c(4)*c(7)*c(13)-c(4)^2*c(6)*c(13)+c(4)^3*c(5)*c(13)-c(4)^2*c(7)+c(4)^3*c(6)-c(4)^4*c(5))$
[24] :

$(-c(7)*c(15)+c(4)*c(6)*c(15)-c(4)^2*c(5)*c(15)+c(4)*c(7)*c(14)-c(4)^2*c(6)*c(14)+c(4)^3*c(5)*c(14)-c(4)^2*c(7)*c(13)+c(4)^3*c(6)*c(13)-c(4)^4*c(5)*c(13)+c(4)^3*c(7)-c(4)^4*c(6)+c(4)^5*c(5))$
[25] :

$(-c(7)*c(16)+c(4)*c(6)*c(16)-c(4)^2*c(5)*c(16)+c(4)*c(7)*c(15)-c(4)^2*c(6)*c(15)+c(4)^3*c(5)*c(15)-c(4)^2*c(7)*c(14)+c(4)^3*c(6)*c(14)-c(4)^4*c(5)*c(14)+c(4)^3*c(7)*c(13)-c(4)^4*c(6)*c(13)+c(4)^5*c(5)*c(13)-c(4)^4*c(7)+c(4)^5*c(6)-c(4)^6*c(5))$
[26] :

$(-c(7)*c(17)+c(4)*c(6)*c(17)-c(4)^2*c(5)*c(17)+c(4)*c(7)*c(16)-c(4)^2*c(6)*c(16)+c(4)^3*c(5)*c(16)-c(4)^2*c(7)*c(15)+c(4)^3*c(6)*c(15)-c(4)^4*c(5)*c(15)+c(4)^3*c(7)*c(14)-c(4)^4*c(6)*c(14)+c(4)^5*c(5)*c(14)-c(4)^4*c(7)*c(13)+c(4)^5*c(6)*c(13)-c(4)^6*c(5)*c(13)-c(9)*c(12)-c(5)^2*c(12)+c(4)^5*c(7)-c(4)^6*c(6)+c(4)^7*c(5))$
[27] :

$(c(5)*c(20)-c(4)*c(5)*c(19)-c(7)*c(18)+c(4)*c(6)*c(18)+c(4)*c(7)*c(17)-$

$c(4)^2 * c(6) * c(17) - c(4)^2 * c(7) * c(16) + c(4)^3 * c(6) * c(16) + c(4)^3 * c(7) * c(15) - c(4)^4 * c(6) * c(15) - c(4)^4 * c(7) * c(14) + c(4)^5 * c(6) * c(14) + c(4)^5 * c(7) * c(13) - c(4)^6 * c(6) * c(13) - c(10) * c(12) - c(5) * c(6) * c(12) - c(4)^6 * c(7) + c(4)^7 * c(6))$

[28]:

$(c(6) * c(20) - c(4) * c(5) * c(20) - c(7) * c(19) + c(4) * c(7) * c(18) - c(4)^2 * c(7) * c(17) + c(4)^3 * c(7) * c(16) - c(4)^4 * c(7) * c(15) + c(4)^5 * c(7) * c(14) - c(4)^6 * c(7) * c(13) - c(11) * c(12) - c(5) * c(7) * c(12) + c(4)^7 * c(7))$

In fact, there are 20 new parameters to insert and the obtained marked polynomials depending on the new variables are $\mathbf{x_3} + (c_1)x_2 + (c_2)x_1 + (c_3)x_0$, $\mathbf{x_1x_2} + (c_4)x_0x_2 + (c_5)x_1^2 + (c_6)x_0x_1 + (c_7)x_0^2$, $\mathbf{x_2^2} + (c_8)x_0x_2 + (c_9)x_1^2 + (c_{10})x_0x_1 + (c_{11})x_0^2$, $\mathbf{x_1^8} + (c_{12})x_0^7x_2 + (c_{13})x_0x_1^7 + (c_{14})x_0^2x_1^6 + (c_{15})x_0^3x_1^5 + (c_{16})x_0^4x_1^4 + (c_{17})x_0^5x_1^3 + (c_{18})x_0^6x_1^2 + (c_{19})x_0^7x_1 + (c_{20})x_0^8$.

The 5 EK-polynomials are $(c_4) * x_0 * x_2 * x_3 + (c_5) * x_1^2 * x_3 + (c_6) * x_0 * x_1 * x_3 + (c_7) * x_0^2 * x_3 + (-c_1) * x_1 * x_2^2 + (-c_2) * x_1^2 * x_2 + (-c_3) * x_0 * x_1 * x_2$

$(c_4) * x_0 * x_2^2 + (c_5) * x_1^2 * x_2 + (-c_8 + c_6) * x_0 * x_1 * x_2 + (c_7) * x_0^2 * x_2 + (-c_9) * x_1^3 + (-c_{10}) * x_0 * x_1^2 + (-c_{11}) * x_0^2 * x_1$

$(c_8) * x_0 * x_2 * x_3 + (c_9) * x_1^2 * x_3 + (c_{10}) * x_0 * x_1 * x_3 + (c_{11}) * x_0^2 * x_3 + (-c_1) * x_2^3 + (-c_2) * x_1 * x_2^2 + (-c_3) * x_0 * x_2^2$

$(c_{12}) * x_0^7 * x_2 * x_3 + (c_{13}) * x_0 * x_1^7 * x_3 + (c_{14}) * x_0^2 * x_1^6 * x_3 + (c_{15}) * x_0^3 * x_1^5 * x_3 + (c_{16}) * x_0^4 * x_1^4 * x_3 + (c_{17}) * x_0^5 * x_1^3 * x_3 + (c_{18}) * x_0^6 * x_1^2 * x_3 + (c_{19}) * x_0^7 * x_1 * x_3 + (c_{20}) * x_0^8 * x_3 + (-c_1) * x_1^8 * x_2 + (-c_2) * x_1^9 + (-c_3) * x_0 * x_1^8$

$(c_{12}) * x_0^7 * x_2^2 + (c_{13} - c_4) * x_0 * x_1^7 * x_2 + (c_{14}) * x_0^2 * x_1^6 * x_2 + (c_{15}) * x_0^3 * x_1^5 * x_2 + (c_{16}) * x_0^4 * x_1^4 * x_2 + (c_{17}) * x_0^5 * x_1^3 * x_2 + (c_{18}) * x_0^6 * x_1^2 * x_2 + (c_{19}) * x_0^7 * x_1 * x_2 + (c_{20}) * x_0^8 * x_2 + (-c_5) * x_1^9 + (-c_6) * x_0 * x_1^8 + (-c_7) * x_0^2 * x_1^7$, from which the above equations can be found.

Example 6.2.20. Consider now a more complicated example.

We type on Singular the following code:

```
LIB"JMSConst.lib";
ring r = 0,  (x(0..5)),rp;
ideal Borid=x(1)^2*x(2),x(0)*x(2)^2,x(1)*x(2)^2,x(2)^3,x(1)^2*x(3),
x(0)*x(2)*x(3),x(1)*x(2)*x(3),x(2)^2*x(3),x(0)*x(3)^2,x(1)*x(3)^2,x(2)*x(3)^2,
x(3)^3,x(1)^2*x(4),x(0)*x(2)*x(4),x(1)*x(2)*x(4),x(2)^2*x(4),
x(0)*x(3)*x(4),x(1)*x(3)*x(4),x(2)*x(3)*x(4),x(3)^2*x(4),x(0)*x(4)^2,
x(1)*x(4)^2,x(2)*x(4)^2,x(3)*x(4)^2,x(4)^3,x(1)^2*x(5),x(0)*x(2)*x(5),
x(1)*x(2)*x(5),x(2)^2*x(5),x(0)*x(3)*x(5),x(1)*x(3)*x(5),x(2)*x(3)*x(5),x(3)^2*
x(5),x(0)*x(4)*x(5),x(1)*x(4)*x(5),x(2)*x(4)*x(5),
x(3)*x(4)*x(5),x(4)^2*x(5),x(0)*x(5)^2,x(1)*x(5)^2,x(2)*x(5)^2,x(3)*x(5)^2,
x(4)*x(5)^2,x(5)^3,x(1)^4;
```

`JMarkedScheme(Borid,r);`
According to the Singular timer function, the 1860 equations resulting for the ideal $J = Borid$ in $\mathbf{k}[x_0, ..., x_5]$ have been computed in about 1 minute and 12 seconds.

To conclude, we point out that the libraries `JMBTest.lib` and `JMSConst.lib` provide solutions only for the *homogeneous case*. As explained in [9], it is possible to work with marked bases and schemes in the non-homogeneous case and so we would like to provide also an implementation in this new setting. A possible application is the problem of smoothability of some local Gorenstein Artin algebras.

6.3 Janet decomposition.

In this section we loosely base on the paper [54], where Janet first defines the notion of *multiplicative variable* for a term τ with respect to a given set $M \subseteq \mathcal{T}$.

For completeness' sake, we recall Janet's decomposition into disjoint classes for terms in the semigroup ideal generated by M.

Each of them contains:

1. a term $\tau \in M$;
2. the set of terms obtained multiplying τ by products of multiplicative variables, that we call *offspring of* τ and denote by $\text{off}_M(\tau)$.

The main difference with respect to Janet's papers is that we remove the finiteness condition on M, showing that it is not necessary for our purposes.

Definition 6.3.1. [54, ppg.75-9] Let $M \subset \mathcal{T}$ be a set of terms and $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ be an element of M. A variable x_j is called *multiplicative* for τ with respect to M if there is no term in M of the form $\tau' = x_1^{\beta_1} \cdots x_j^{\beta_j} x_{j+1}^{\alpha_{j+1}} \cdots x_n^{\alpha_n}$ with $\beta_j > \alpha_j$. We will denote by $mult_M(\tau)$ the set of multiplicative variables for τ with respect to M.

Definition 6.3.2. With the previous notation, the *offspring* of τ with respect to M is the set

$$\text{off}_M(\tau) := \{\tau x_1^{\lambda_1} \cdots x_n^{\lambda_n} \mid \text{where } \lambda_j \neq 0 \text{ only if } x_j \text{ is multiplicative for } \tau \text{ w.r.t. } M\}.$$

Example 6.3.3. Consider the set $M = \{x_1^3, x_2^3, x_1^4 x_2 x_3, x_3^2\} \subseteq \mathbf{k}[x_1, x_2, x_3]$.
Let $\tau = x_1^3$, so $\alpha_1 = 3$, $\alpha_2 = \alpha_3 = 0$. The variable x_1 is multiplicative for τ w.r.t M since there are no terms $\tau' = x_1^{\beta_1} x_2^{\beta_2} x_3^{\beta_3} \in M$ satisfying both conditions:

- $\beta_1 > 3$;

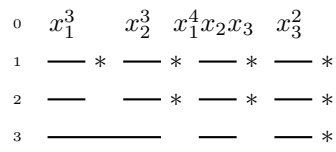
- $\beta_2 = \beta_3 = 0$.

On the other hand, x_2 is not multiplicative for τ since $\tau'' = x_2^3 \in M$ satisfies
$\tau'' = x_1^{\gamma_1} x_2^{\gamma_2} x_3^{\gamma_3}$ with $\gamma_2 = 3 > 0 = \alpha_2$, $\gamma_3 = \alpha_3 = 0$.
Similarly, x_3 is not multiplicative since $x_3^2 \in M$.
In conclusion, we have $mult_M(\tau) = \{x_1\}$.

Remark 6.3.4. Observe that, by definition of multiplicative variable, the only element in $\text{off}_M(\tau) \cap M$ is τ itself.
Indeed, if $\tau \in M$ and also $\tau\sigma \in M$ for a non constant term σ, then $max(\sigma)$ cannot be multiplicative for τ, hence $\tau\sigma \notin \text{off}_M(\tau)$.

Given a finite set of terms $M \subseteq \mathcal{T}$, we can easily list the multiplicative variables of its elements by a Bar Code construction.
More precisely, let B_M the Bar Code associated to M, as defined in 5.2.5. After drawing B_M, we place the stars in the diagram as for the star set computation, obtaining the Bar Code picture (c.f. section 5.3).
Let A be an i-bar, followed by a star. Then, for all $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in M$ lying over A, $x_i \in mult(\tau)$.
Indeed, if $i = n$, $\nexists \sigma \in M$ such that $deg_n(\sigma) > \alpha_n$ because, if there was such a σ, by hypothesis, being $\sigma >_{Lex} \tau$, it would lie over a n-bar posed on the right of A so, by construction, A would not be followed by a star.
On the other hand, if $i < n$, let B be the $(i+1)$-bar over which A lies. The bar A is followed by a star so, as explained in section 5.3, also B interrupts in correspondence of the end of the bar A.
If x_i was non multiplicative for τ then $\exists \sigma = x_1^{\beta_1} \cdots x_i^{\beta_i} x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n}$ with $\beta_i > \alpha_i$. The term σ would lie over B ($deg_{i+1}(\sigma) = \alpha_{i+1} = deg_{i+1}(\tau)$) but it would be over an i-bar A', posed on the right of A over B, which cannot exist by the procedure to set the stars.
Let now $\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in M$ and let $x_i \in mult(\tau)$. We prove that the i-bar A underlying τ is followed by a star.
As done for the comments above, we denote by B the $(i+1)$-bar over which the bar A lies. If A is not followed by a star, B does not interrupt in correspondence to the end of A, so there is an i-bar A' over B and posed on the right of A.
If $\sigma \in M$ is a term lying over A', $deg_i(\sigma) = \alpha_i + 1$, $deg_{i+1}(\sigma) = \alpha_{i+1}$,...,$deg_n(\sigma) = \alpha_n$ and so, by definition 6.3.1, x_i is not multiplicative for τ.

Example 6.3.5. For the set $M = \{x_1^3, x_2^3, x_1^4 x_2 x_3, x_3^2\} \subseteq \mathbf{k}[x_1, x_2, x_3]$ of example 6.3.3, we have the following Bar Code picture



Then:

- $mult(x_1^3) = \{x_1\}$;
- $mult(x_2^3) = \{x_1x_2\}$;
- $mult(x_1^4x_2x_3) = \{x_1, x_2\}$;
- $mult(x_3^2) = \{x_1, x_2, x_3\}$.

In paper [54], Janet defining multiplicative variables as in Definition 6.3.1, provides both a decomposition for the semigroup ideal $\mathsf{T}(M)$ generated by a finite set of terms M and a decomposition for the complementary set $\mathsf{N}(M)$.
On the other hand, in [55, 56], he defines multiplicative variables in the following way.

6.3.6. A variable x_j is *multiplicative* for $\tau \in \mathcal{T}$ if and only if $x_j \leq \min(\tau)$.

We denote by mult the multiplicative variables in this sense.
These two definitions of multiplicative variables appear to be very different.
First of all, in the first formulation, the set of multiplicative variables for a term in M depends on the whole set M, while in the second it is completely independent on the set M.
Indeed, the two notions are not equivalent for a general set M, as shown by the following examples.

Example 6.3.7. In $\mathbf{k}[x_1, x_2, x_3]$ consider the ideal $I = (x_1^2x_2, x_1x_2^2)$ and let M be its monomial basis. Then, $mult_M(x_1^2x_2) = \{x_1, x_3\}$ and $mult_M(x_1x_2^2) = \{x_1, x_2, x_3\}$ while only x_1 can be multiplicative according to the other notion of multiplicative variable.

Example 6.3.8. Taken the set $M = \{x_1^2x_2, x_1x_2^2\} \subseteq \mathbf{k}[x_1, x_2]$, we get $mult_M(x_1x_2^2) = \{x_1, x_2\}$, while of course $x_1 \leq \min(x_1x_2^2)$ but $x_2 > \min(x_1x_2^2)$.

However, they are equivalent in Janet setting, that is if M is the generating set of the generic initial ideal of homogeneous ideals I.
More generally, we will see that they turn out to be equivalent also if M is the monomial basis $\mathsf{G}(J)$ of a strongly stable ideal J and if M is the special set of generators of any monomial ideal J denoted by $\mathcal{F}(J)$ (see 5.3 and 6.4).

We will see that stronger results can be proved when a set M is such that *the two definitions* of multiplicative variables *coincide*.

The following definition is a key point for this chapter.

Definition 6.3.9. [54, ppg.75-9] A set of terms $M \subset \mathcal{T}$ is called *complete* if for every $\tau \in M$ and $x_j \notin mult_M(\tau)$, there exists $\tau' \in M$ such that $x_j\tau \in \text{off}_M(\tau')$.

Moreover, M is *stably complete* if it is complete and for every $\tau \in M$ it holds $mult_M(\tau) = \{x_i \mid x_i \leq \min(\tau)\}$.
If a set M is stably complete and finite, then it is the *Pommaret basis* of $J = (M)$ and we denote it by $\mathcal{H}(J)$.

Remark 6.3.10. If $M = \{\tau\} \subseteq \mathcal{Q}$ is a singleton, it is complete, with $mult(\tau) = \{x_1, ..., x_n\}$.

Let us examine some examples.

Example 6.3.11. In $\mathbf{k}[x_1, x_2, x_3]$ consider the ideal $I = (x_1^2, x_1x_2, x_3)$.
Both $M_0 = \{x_1^2, x_1x_2, x_3\}$ and each generating set of I with the shape
$M_i = \{x_1^2, x_1x_2, x_3, x_2x_3, ..., x_2^ix_3\}$ are complete systems of terms. In fact, for M_0:

- $mult_{M_0}(x_1^2) = \{x_1\}$, $x_1^2x_2 \in \text{off}_{M_0}(x_1x_2)$, $x_1^2x_3 \in \text{off}_{M_0}(x_3)$;
- $mult_{M_0}(x_1x_2) = \{x_1, x_2\}$, $x_1x_2x_3 \in \text{off}_{M_0}(x_3)$;
- $mult_{M_0}(x_3) = \{x_1, x_2, x_3\}$.

For M_i, $i \geq 1$:

- $mult_{M_i}(x_1^2) = \{x_1\}$, $x_1^2x_2 \in \text{off}_{M_i}(x_1x_2)$, $x_1^2x_3 \in \text{off}_{M_i}(x_3)$;
- $mult_{M_i}(x_1x_2) = \{x_1, x_2\}$, $x_1x_2x_3 \in \text{off}_{M_i}(x_2x_3)$;
- $mult_{M_i}(x_3) = \{x_1, x_3\}$, $x_2x_3 \in \text{off}_{M_i}(x_2x_3)$;
- $mult_{M_i}(x_2^jx_3) = \{x_1, x_3\}$, $x_2^{j+1}x_3 \in \text{off}_{M_i}(x_2^{j+1}x_3)$, $0 \leq j < i$;
- $mult_{M_i}(x_2^ix_3) = \{x_1, x_2, x_3\}$.

Example 6.3.12. Consider the ideal $J = (xy) \lhd \mathbf{k}[x, y]$.
The monomial basis $M_0 = \mathsf{G}(J) = \{xy\}$ is a complete system with $mult_{M_0}(xy) = \{x, y\}$.
Also the set $M = \{x^hy \mid h \geq 1\} \subseteq \mathbf{k}[x, y]$, $x < y$, is a complete system, again according to the first definition. It generates the same ideal (xy), but has infinitely many elements. Anyway, it is not stably complete. In fact, for each $x^hy \in M$, $mult_M(x^hy) = \{y\}$, since no terms of the form x^ly^e with $e > 1$ belong to M; on the other hand $x \notin mult_M(x^hy)$ since $x^{h+1}y \in M$.

Example 6.3.13. Let M be the set of terms $\{x, y^2\}$ in $\mathbf{k}[x, y]$, with $x < y$.
The multiplicative variables for every term in M are those lower than or equal to its minimal one:
mult$(x) = \{x\}$ mult$(y^2) = \{x, y\}$.
However, M is not complete since yx does not belong to the offspring of any term in M.

The following example shows that a complete generating set of terms can loose completeness when the ideal is enlarged.

Example 6.3.14. Let $M = \{x^2, xy\} \subset \mathbf{k}[x, y]$ and $J = (M)$. It is a complete system, but it is not stably complete, since y is multiplicative for xy, although $\min(xy) = x$.
Adding to M a term in $\mathsf{N}(J)$, we get a new set M_0 and $J_0 = (M_0)$, whose Janet decomposition clearly changes. For example, if $M_0 = \{x^2, xy, y^2\}$ we get a stably complete system.
On the other hand, if $M_0 = \{x^2, xy, y^3\}$ the system is not complete anymore, since xy^2 does not belong to the offspring of any term in the set.

From definition 6.3.1 of multiplicative variable, Janet deduces the following straightforward corollary

Corollary 6.3.15 ([54])**.** Let $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$ be a finite set of terms, $\tau_i = x_1^{\alpha_1^{(i)}} \cdots x_n^{\alpha_n^{(i)}}$ and $\tau_i' = x_1^{\alpha_1^{(i)}} \cdots x_{n-1}^{\alpha_{n-1}^{(i)}} = \frac{\tau_i}{x_n^{deg_n(\tau_i)}}$, for $i = 1, ..., m$.
Let $D_n := \{\beta \in \mathbb{N} | \exists \tau \in M,\, deg_n(\tau) = \beta\}$, $\alpha^{(n)} := \max(D)$ and, for each $\beta \in D_n$, define $M'_\beta := \{\frac{\tau}{x_n^{deg_n(\tau)}},\, \tau \in M \text{ and } deg_n(\tau) = \beta\}$.
Then M is complete if and only if the two conditions below hold:

1. $\forall \beta \in D_n$, M'_β is a complete set;
2. $\forall \tau'_i \in M'_\beta$, $\beta < \alpha^{(n)}$ there exists $j \in \{1, ..., m\}$ such that
   - $\tau'_i \in$ off(τ'_j);
   - $\tau'_j \in M'_{\beta+1}$.

Completeness of a given finite set M can be detected by exploiting the Bar Code structure.
If $\tau \in M$ and $x_i \notin mult(\tau)$, let A be the i-bar underlying τ and A' the subsequent i-bar[7].
If, $\forall \sigma$ over A', $\sigma \nmid x_i\tau$, the system is not complete.
If $\exists \sigma$ over A', $\sigma \mid x_i\tau$, so that $x_i\tau = \sigma\eta$, let $V := \{x_j,\, 1 \leq j \leq n,\, x_j \mid \eta\}$, the set of the variables appearing in η with nonzero exponent. If, for each $x_j \in V$, the j-bar underlying σ

[7]We recall that the *last* term in M is $\xi := \max_{Lex}(M)$ and, by the comments above, $mult(\xi) = \{x_1, ..., x_n\}$.

is followed by a star, then $\tau \in \text{off}(\sigma)$ and we continue examining the next term in M.
If there exists a variable $x_j \in V$ such that the the j-bar underlying σ is not followed by a star, then the system is *not* complete.
First of all, we explain why we look for σ *only over* A' and not over other i-bars.

- $x_i\tau \notin \text{off}(\sigma)$ for σ lying over A, since A is not followed by a star.
- Let A'' be an i-bar posed on the *right* of A'. If σ lies over A'', $x_i\tau \notin \text{off}(\sigma)$ since $\sigma \nmid \tau$, being $deg_j(\sigma) > deg_j(x_i\tau)$ for some $j \geq i$.
- Let A''' be an i-bar posed on the left of A. If $\sigma = x_1^{\beta_1} \cdots x_n^{\beta_n}$ lies over A''', then $\sigma <_{Lex} \tau$ and σ cannot be such that $deg_j(\sigma) = \alpha_j$ for $j = i, ..., n$ because, if it is like that, it would lay over A.
  This implies that if $x_j = \max\{x_h,\, h = 1, ..., n,\, deg_h(\sigma) < \deg_h(\tau)\}$, then $x_j \notin mult(\sigma)$:

$$\tau = x_1^{\alpha_1} \cdots x_n^{\alpha_n},\ \alpha_j > \beta_j,\ \alpha_{j+1} = \beta_{j+1}, ..., \beta_n = \alpha_n$$

Now, let $\tau \in M$, $x_1 \in mult(\tau)$, A the i-bar underlying τ and A' the subsequent i-bar.
If it is possible to find a $\sigma \mid x_i\tau$ lying over A', with all the bars lying under σ and corresponding to the variables of $\eta := \frac{x_i\tau}{\sigma}$ followed by stars, then $\tau \in \text{off}(\sigma)$. It is clear since the variables in η turn out to be multiplicative for τ.
On the other hand, if $\tau \in \text{off}(\sigma)$, the bars underlying σ, which correspond to the variables of $\eta := \frac{x_i\tau}{\sigma}$ are followed by stars. Indeed, η is composed by multiplicative variables for σ.

Another problem one can pose is:

Problem 6.3.16. Given a finite set of terms $M = \{\tau_1, ..., \tau_m\} \subseteq \mathcal{T}$ is there any *ordering on the variables* $x_1, ..., x_n$ such that M is *complete*?

As explained above, the Bar Code construction allows to detect the completeness of M. Clearly such a construction depends on the variables' ordering, so if we want to solve problem 6.3.16 we should draw and check $n!$ different Bar Codes, which turns out to be rather tedious and time consuming.
Exploiting again the Bar Code structure and corollary 6.3.15, we can look for the solution of 6.3.16 in a "greedy" way, so that most of the tests can be skipped.
More precisely, considered $M = \{\tau_1, ..., \tau_m\}$, we perform the steps described below.

Step a) *Quest for the maximal variable.*
Let C be the set containing all the candidates for being the maximal variable in the

ordering we are going to construct.
A priori, all the variables can be good candidates for the role of maximal variable, so $C = \{x_1, ..., x_n\}$. It is necessary to examine the variables, in order to estabilish which of them can really hold this position, in order to have a complete system.

1. For $i = 1, ..., n$, compute the sets

$$D_i := \{\beta \in \mathbb{N} | \exists \tau \in M,\, deg_i(\tau) = \beta\}.$$

2. Read each of these sets: if, for some $1 \leq j \leq n$, there are two $\gamma_1, \gamma_2 \in D_j$, $\gamma_1 < \gamma_2 - 1$ and $\exists \gamma_3$, $\gamma_1 < \gamma_3 < \gamma_2$, such that $\gamma_3 \notin D_j$, then x_j cannot be the maximal variable for our ordering. Indeed, if so, $M'_{\gamma_3} = \emptyset$ and this contradicts corollary 6.3.15. In this case, exclude x_j from C.

Test) If $C = \emptyset$, none of the variables can be the maximal one and this implies that the system is not complete for any variable ordering. Otherwise, we continue.

Choice) Pick an element $x_i \in C \neq \emptyset$ which we assume to be the maximal variable for the ordering we are constructing. Then set $C = C \setminus \{x_i\}$[8].

Step b) *Divisors and multiplicative variables.*

1. Write down the terms in M, arranging them w.r.t. their i-degree. If, for some $\tau_{j1}, \tau_{j2} \in M$, $deg_i(\tau_{j1}) = deg_i(\tau_{j2})$ and $\tau_{j1} \mid \tau_{j2}$, then write τ_{j1} on the left of τ_{j2}. This operation is equivalent to draw the lowest line of the Bar Code associated to the variable ordering we are creating step by step[9]. From now on, we denote by $A^{(i)}_{-}$ these bars. We encode them, together with the terms.
2. For each τ_{j1}, lying over $A^{(i)}_1$, check whether there are terms τ_{j2} over $A^{(i)}_2$ such that $deg_h(\tau_{j2}) \leq deg_h(\tau_{j1})$, for each $h \neq i$. Do the same for the couples of consecutive i-bars $A^{(i)}_2$, $A^{(i)}_3$; ...;$A^{(i)}_{\mu(i)-1}$, $A^{(i)}_{\mu(i)}$.
   - If the quest has positive outcome, we keep track of the terms we found, together with all the variables $h \neq i$ for which the strict inequality holds. These are those we need to belong to mult(τ_{j2}) so that $x_i\tau_{j1} \in$ off(τ_{j2}).
   - If, for some term τ there are no σ satisfying the properties described above, the test fails[10]. So we break.

[8] This means that we examine x_i as maximal variable only once.
[9] If x_i is the maximal variable, for each $\tau \in M$, $P_{x_i}(\tau) = x_i^{deg_i(\tau)}$, so each i-bar identifies an i-degree.
[10] Actually, this means that, in corollary 6.3.15, point 2. is not verified.

Test 2) If Step b) reports a failure, delete the bars and go to Test)[11]. Otherwise continue.

Once a variable has been selected, we have to choose the following one in order of magnitude.
We have as information, the Bar Code already drawn, the variable already settled, together with the list of the other possible candidates for the whole set of positions under consideration at this point and the list of variables we need to be multiplicative for each term. Moreover, if we are not dealing with the second variable in order of magniture, we have some information on the previous variables, namely for which terms they are multiplicative.
We have to repeat what follows for settling down the other variables, until we get either an ordering (i.e. we have settled all the variables, so that we can quit with a positive outcome) or a situation in which all the bars are unitary (we will explain this situation below).

Ordering) If all the variables have been settled, we have found an ordering on the variables, for which M is complete, so we quit with positive outcome. Otherwise we continue with the next step.

Unitary) If all the examined bars are unitary[12], we can quit with a positive outcome since the ordering on the other variables, not already examined, is indifferent. Otherwise we continue with the next step.

Step c) *Next variable.*

Candidates) For each bar $A_h^{(j)}$, $h = 1, ..., \mu(j)$, on the topmost line already drawn, there is a set of terms lying over it.
Execute Step a) over each of these sets (forgetting about the variables already settled) and intersect the obtained sets of candidates.

Test 3) If such an intersection is empty, we have to come back. More precisely we delete the topmost line in the Bar Code (and the related information, except that the candidate list). Then, if there are no bar left, we go to Test), otherwise, we repeat Test 3) on the set of candidates related to the variable treated in the previous step [13]. If it is nonempty, we continue with the following step.

Pick) Select an element x_l from the list of candidates found in Candidates) and delete it from the list.

[11] i.e. we change the candidate maximal variable: M cannot be complete w.r.t. any variable ordering with maximal variable x_i.
[12] Notice that singletons are complete. Moreover, all the variables not already settled have to be multiplicative.
[13] We are selecting another element in the list of candidates for the variable treated in the previous step.

Step b′) *Divisors and multiplicative variables.* Since this step is analogous to Step b), we only sketch it, referring to Step b).

1. Order the terms over each bar w.r.t. their degree on the new candidate. If, for some $\tau_{j1}, \tau_{j2} \in M$, $deg_i(\tau_{j1}) = deg_i(\tau_{j2})$ and $\tau_{j1} \mid \tau_{j2}$, then write τ_{j1} on the left of τ_{j2}. Draw the associated bars.
2. Repeat the same test as in Step b) 2 for all the couples of consecutive bars lying over the same one w.r.t. the variable treated in the previous step. Keep track of the terms or report failure and break as in Step b) 2.

Test 2) If Step b′) reports a failure, delete the upmost line of the Bar Code and go back to Test 3). Otherwise continue with next step.

Compatibility) The new candidate x_l has to be compatible with the variables chosen so far.

1. Read the terms of which we have kept track, together with the variables we need to be multiplicative, in order to have a complete set of terms.
2. If x_l is one of the variables associated to some τ, check whether is multiplicative or not. This means looking whether τ lies on the rightmost l-bar over the underlying bar (in this case x_l is multiplicative for τ) or not. Notice that a negative outcome do not authomatically exclude the completeness of the system, since a term could potentially have associated to it more than one term arising from Steps b), b′), equipped with some variables, required to be multiplicative.
   - If x_l is not multiplicative for some τ, but τ is not the only term we have recorded for the term under consideration, we mark τ as failed w.r.t x_l[14].
   - If, for some term in M, all the associated terms we have kept track of give a negative outcome, the test fails[15]. So we break.

Test 4) If the Compatibility) fails, we change candidate for the current variable, i.e. we reset the markers set for x_l, we delete the upmost line in the Bar Code and we go to Test 3). Otherwise we quit Step c).

Redirection) Go to Ordering).

Example 6.3.17. Consider $M = \{x_1x_2^3, x_1^3x_2\} \subset \mathbf{k}[x_1, x_2]$. Such a set is not complete by a) since $D_1 = D_2 = \{1, 3\}$.
As a confirmation, we can see that, if $x_1 < x_2$, we have

[14] If the test passes, the failed terms are not examined anymore.
[15] The first condition of 6.3.15 is not verified.

$$\begin{array}{ll} {\scriptstyle 0} & x_1^3x_2 \quad x_1x_2^3 \\ {\scriptstyle 1} & \text{—} * \text{—} * \\ {\scriptstyle 2} & \text{—} \quad \text{—} * \end{array}$$

Then $mult(x_1^3x_2) = \{x_1\}$, $mult(x_1x_2^3) = \{x_1, x_2\}$ and $x_1^3x_2^2$ does not belong either to off$(x_1^3x_2)$ or to off$(x_1x_2^3)$.
On the other hand, if $x_2 < x_1$, we have

$$\begin{array}{ll} {\scriptstyle 0} & x_1x_2^3 \quad x_1^3x_2 \\ {\scriptstyle 2} & \text{—} * \text{—} * \\ {\scriptstyle 1} & \text{—} \quad \text{—} * \end{array}$$

Thus $mult(x_1x_2^3) = \{x_2\}$, $mult(x_1^3x_2) = \{x_1, x_2\}$ and $x_1^2x_2^3$ does not belong either to off$(x_1^3x_2)$ or to off$(x_1x_2^3)$.

Example 6.3.18. Consider

$$M = \{x_2x_3, x_1^2, x_3^2, x_2^2, x_1x_2, x_1x_2x_4, x_1^2x_4, x_4x_3, x_2^2x_4, x_1^2x_3\} \subset \mathbf{k}[x_1, x_2, x_3, x_4].$$

Step a) $D_1 = D_2 = D_3 = \{0, 1, 2\}$, $D_4 = \{0, 1\}$.
Each variable is a good candidate for being the maximal one, so we move to Step b), choosing, for example, x_3, getting

$$\begin{array}{lllllllll} & x_1^2 & x_1x_2 & x_2^2 & x_1^2x_4 & x_1x_2x_4 & x_2^2x_4 & x_1^2x_3 & x_2x_3 \quad x_4x_3 \quad x_3^2 \\ {\scriptstyle 3} & \text{———} & \text{———} & \text{———} & \text{———} & \text{———} & \text{———} & \text{———} & \text{———} \quad \text{———} \quad \text{—} \end{array}$$

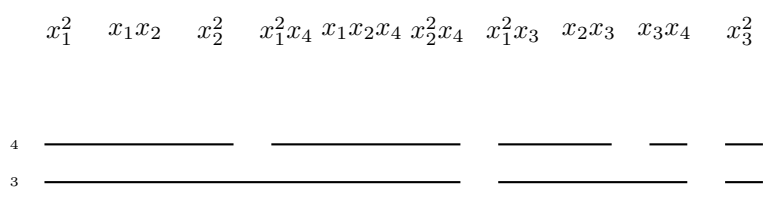
For $A_1^{(3)}, A_2^{(3)}, A_3^{(3)}$ we test the divisors.
For $A_1^{(3)}, A_2^{(3)}$:

- $x_1^2 \to x_1^2x_3$. We do not keep track of any variable.
- $x_1x_2 \to x_2x_3$. We keep track of x_1.
- $x_2^2 \to x_2x_3$. We keep track of x_2.
- $x_1^2x_4 \to x_1^2x_3$. We keep track of x_4. $x_1^2x_4 \to x_3x_4$. We keep track of x_1.
- $x_1x_2x_4 \to x_2x_3$. We keep track of x_1, x_4. $x_1x_2x_4 \to x_3x_4$. We keep track of x_1, x_2.
- $x_2^2x_4 \to x_2x_3$. We keep track of x_2, x_4. $x_2^2x_4 \to x_3x_4$. We keep track of x_2.

For $A_2^{(3)}, A_3^{(3)}$:

- $x_1^2x_3 \to x_3^2$. We keep track of x_1.
- $x_2x_3 \to x_3^2$. We keep track of x_2.
- $x_3x_4 \to x_3^2$. We keep track of x_4.

We do not have a negative outcome for any term, so we continue with Step c). Being $A_3^{(3)}$ overlied only by x_3^2 (only one term!) we do not need to take this bar into account (all the variables different from x_3 are good candidates!).
All the variables are good candidates for being the second in order of magnitude and, for example, we choose x_4, getting:



We check the divisors (forgetting about x_3):

- $x_1^2 \to x_1^2x_4$. We do not keep track of any variable.
- $x_1x_2 \to x_1x_2x_4$. We do not keep track of any variable.
- $x_2^2 \to x_2^2x_4$. We do not keep track of any variable.

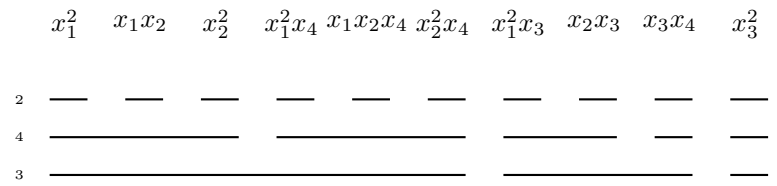
and

- $x_1^2 \to x_4$. We keep track of x_1.
- $x_2 \to x_4$. We keep track of x_2.

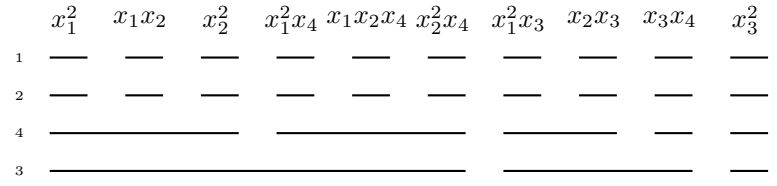
Then we pass to Compatibility):

- $x_1^2x_4$: $x_1^2x_3$ does not lie on the rightmost 4-bar, so x_4 is not multiplicative. Since we have more than one term associated to $x_1^2x_4$, we only delete $x_1^2x_3$ and keep x_3x_4.
  The same reasoning holds for $x_1x_2x_4, x_2^2x_4$.
- x_3x_4: x_3^2 lies on the rightmost 4-bar so it passes the test.

We continue choosing x_2 as next variable and we get:



This way, all the sets overlying the 2-bars are singletons. We check on the 2-bars to have nonincreasing exponents for x_1 and this is true. Moreover, we check that x_2 is multiplicative where it is marked, i.e. for x_2x_3, x_3x_4 but it clearly holds.
The system M is complete for $x_1 < x_2 < x_4 < x_3$ and its final Bar Code w.r.t. the chosen ordering is



The following technical lemma will be very useful throughout the paper. As a first application, we will prove (theorem 6.3.20) that a system of terms M (possibly infinite) is *complete* if and only if the offsprings of the elements in M form a partition of the semigroup ideal generated by M.

Lemma 6.3.19. [57, pg.23] Let τ, τ' be elements of a set of terms M and x_j be a variable such that $x_j \notin \text{mult}_M(\tau)$ and $x_j\tau \in \text{off}_M(\tau')$. Then $\tau <_{Lex} \tau'$. If, moreover, $x_j \leq min(\tau)$, then $\tau x_j = \tau' \in M$.

Proof. First of all, we observe that $\tau \neq \tau'$, since $x_j \notin mult_M(\tau)$. By definition of offspring, we have that $\tau x_j = \tau'\sigma'$, where σ' is a product of multiplicative variables for τ'. Let us assume by contradiction that $\tau >_{Lex} \tau'$ and let x_i be the maximal variable such that $deg_i(\tau) > deg_i(\tau')$. Then, $x_i|\sigma'$, hence $x_i \in mult_M(\tau')$, but this is impossible by definition of multiplicative variable, since also τ is in M.

Now let us assume that $x_j \leq min(\tau)$ and $\sigma' \neq 1$. If $x_j|\sigma'$, then $\tau = \frac{\sigma'}{x_j}\tau' \in M \cap \text{off}_M(\tau')$, which is not possible by Remark 6.3.4. If, on the contrary, $x_j \not| \sigma'$ we get a contradiction with the previous assertion, since in this case $\tau' \leq_{Lex} \frac{\tau'\sigma'}{\max(\sigma')} <_{Lex} \frac{\tau'\sigma'}{x_j} = \tau$. □

Theorem 6.3.20. Let M be a set of terms (possibly infinite).

If $\tau, \tau' \in M$ and $\tau \neq \tau'$, then $\text{off}_M(\tau) \cap \text{off}_M(\tau') = \emptyset$.

If, moreover, M is complete and $\mathsf{T}(M)$ is the semigroup ideal it generates, then $\forall \gamma \in \mathsf{T}(M)$, $\exists \tau \in M$ such that $\gamma \in \mathrm{off}_M(\tau)$. Hence, the offsprings of the elements in M give a partition of $\mathsf{T}(M)$.

Proof. To prove the first assertion, let us assume by contradiction that $\tau\sigma = \tau'\sigma' \in \mathrm{off}_M(\tau) \cap \mathrm{off}_M(\tau') \neq \emptyset$ and let $\tau >_{lex} \tau'$. If x_i is the maximal variable such that $deg_i(\tau) > deg_i(\tau')$, then $x_i|\sigma'$. By definition of offspring, $x_i \in mult_M(\tau')$, but this is impossible by definition of multiplicative variable, since also τ is in M.

Now we assume that M is complete and prove the second fact. We argue by contradiction. Suppose $\mathsf{T}(M) \supsetneq O := \bigcup_{\sigma \in M} \mathrm{off}_M(\sigma)$ and take any term γ in $\mathsf{T}(M) \setminus O$. As M generates $\mathsf{T}(M)$, there are terms in M that divide γ: let τ be the one which is maximal with respect to $<_{lex}$. If $\gamma = \tau\sigma$, the term σ contains at least a variable x_i which is not multiplicative for τ, since $\tau\sigma \notin \mathrm{off}_M(\tau)$. Then $\gamma = \tau x_i \eta$ and $\tau x_i \notin \mathrm{off}_M(\tau)$.

By the completeness of M, we have $\tau x_i \in O$, namely there is a term $\tau' \in M$ such that $\tau x_i = \tau'\sigma' \in \mathrm{off}_M(\tau')$. By Lemma 6.3.19 *i)*, $\tau' >_{Lex} \tau$, and this is not possible since $\tau'|\gamma = \tau x_i \eta = \tau'\sigma'\eta$. □

Thanks to the previous result, if M is a complete system, each term in $\mathsf{T}(M)$ can be written in a unique way as a product of

1. an element $\tau \in M$;
2. a term $x^\eta = x_i^{\eta_i} \cdots x_j^{\eta_j}$, with $x_i, ..., x_j \in \mathrm{mult}_M(\tau)$.

This fact suggests the following

Definition 6.3.21. Let M be a complete system of terms. The *star decomposition* of every term $\gamma \in (M)$ with respect to M, is the unique couple of terms (τ, η), with $\tau \in M$, such that $\gamma = \tau\eta$ and $\gamma \in \mathrm{off}_M(\tau)$. If (τ, η) is the star decomposition of γ with respect to M, we will write $\gamma = \tau *_M \eta$.

Remark 6.3.22. From the results stated above, we obtain the following explicit formula for the Hilbert function of (M):

$$ {}^hHF_{(M)}(k) = \binom{k+n}{n} - \sum_{\tau \in M \; deg(\tau) \leq k} \binom{k - deg(\tau) + s_\tau - 1}{s_\tau - 1}, $$

where s_τ is the number of multiplicative variables for τ w.r.t M and we set equal to 0 every binomial with a negative numerator or a negative denominator.

Thus, this formula makes sense also for infinite sets M, since for every k there are only finitely many non-zero summands.

If M is a finite set of terms and r is the maximal degree of its elements, this formula gives the value of the Hilbert polynomial for every $k \geq r$.

The following lemma will be very useful for the reduction process we will define in section 6.5.

Lemma 6.3.23. Let M be a stably complete system of terms and let γ be a term such that $\gamma = \tau *_M \eta$ and also $\gamma = \sigma\eta'$ with $\sigma \notin \mathsf{T}(M)$.
Then $\eta' >_{Lex} \eta$.

Proof. By definition of stable completeness, $\min(\tau) \geq \max(\eta)$. If $\eta' <_{Lex} \eta$, then $\eta'|\eta$ and $\tau|\sigma$. This is not possible since $\tau \in \mathsf{T}(M)$ and $\sigma \notin \mathsf{T}(M)$. □

6.4 Star set and quasi stable ideals

In this section, we take again under consideration the *star set* of a given monomial ideal $J \lhd P$:

$$\mathcal{F}(J) := \{x^\alpha \in \mathcal{T} \setminus \mathsf{N}(J) \,|\, \frac{x^\alpha}{\min(x^\alpha)} \in \mathsf{N}(J)\}.$$

We will prove that it is a complete system with many interesting properties in common with the minimal monomial basis of strongly stable ideals.

Theorem 6.4.1. For every monomial ideal J, the star set $\mathcal{F}(J)$ is the unique stably complete system of generators of J. Hence, if M is stably complete, $M = \mathcal{F}((M))$.

Proof. Let $\tau := x_k^{\alpha_k} \cdots x_n^{\alpha_n}$ be any monomial in $\mathcal{F}(J)$.

Assume x_i is not multiplicative, so that $x_i\tau \in J$, $x_i\tau = \tau'\sigma', \tau' \in M$. Then Lemma 6.3.19 implies $\tau <_{Lex} \tau'$ whence $x_i > \min(\tau)$.

Let $x_i > x_k := min(\tau)$ and set $\sigma_0 := \tau x_i$, $\sigma_r := \frac{\sigma_{r-1}}{min(\sigma_{r-1})}$ for $r = 1 \ldots, \alpha_k + \cdots + \alpha_{i-1}$. Note that $x_i^{\alpha_i} \cdots x_n^{\alpha_n} \notin J$, since it divides $\frac{\tau}{min(\tau)} \in \mathsf{N}(J)$, while $\sigma := \sigma_0 \in J$, since it is a multiple of τ. Then, in the sequence of terms $\sigma_i, 0 \leq i \leq \alpha_k + \cdots + \alpha_{i-1}$, we find an element σ_j that belongs to J, while the following one does not.

Then $\sigma_j \in \mathcal{F}(J)$, so that $x_i\tau \in \mathrm{off}_{\mathcal{F}(J)}(\sigma_j)$ and x_i is not multiplicative for τ w.r.t. $\mathcal{F}(J)$.

Take $\tau = x_k^{\alpha_k} \cdots x_n^{\alpha_n} \in \mathcal{F}(J)$, and a variable $x_i \notin mult_{\mathcal{F}(J)}(\tau)$. By the previous result $x_i > x_k = \min(\tau)$. By definition of non-multiplicative variable, there is a term $\sigma' = x_i^t\, x_{i+1}^{\alpha_{i+1}} \cdots x_n^{\alpha_n} \in \mathcal{F}(J)$, for some integer $t > \alpha_i$. Let us consider the minimum one.

If $t = \alpha_i + 1$, then $x_i\tau = x_k^{\alpha_k} \cdots x_i^t \cdots x_n^{\alpha_n} \in \mathrm{off}_{\mathcal{F}(J)}(\sigma')$.

If, on the contrary, $t > \alpha_i + 1$, then $\sigma'' = x_i^{\alpha_i+1} \cdots x_n^{\alpha_n} \in \mathsf{N}(J)$ by definition. Let us consider, as in the previous proof, the sequence of terms $\sigma_0 := \tau x_i \in J$, $\sigma_r := \frac{\sigma_{r-1}}{min(\sigma_{r-1})}$ for

$r = 1 \ldots, \sum_{j=k}^{k-1} \alpha_j$. Since the last one is σ'', we can find in this sequence a suitable $\sigma_j \in I$ such that $\sigma_{j+1} \in \mathsf{N}(J)$, that is $\sigma_j \in \mathcal{F}(J)$ and $x_i\tau \in \mathrm{off}_{\mathcal{F}(J)}(\sigma_j)$.

In order to prove that every stably complete set of terms M, with $J = (M)$ is exactly $\mathcal{F}(J)$, we first notice that clearly $\mathsf{G}(J) \subseteq M$ and $\mathsf{G}(J) \subseteq \mathcal{F}(J)$.
Moreover, it is sufficient to prove that $\mathcal{F}(J) \subseteq M$. Let $\sigma \in \mathcal{F}(J)$, i.e. $\frac{\sigma}{\min(\sigma)} = \omega \in \mathsf{N}(J)$. Then, there exists $\tau \in M$ such that $\sigma \in \mathrm{off}(\tau)$ and so $\sigma = \tau\eta$, with either $\eta = 1$ or $\max(\eta) \leq \min(\tau)$.
This implies that either $\tau = \sigma$ or $\tau \mid \omega$, but the second alternative is impossible since both $\tau \in M$ and $\omega \in \mathsf{N}(J)$.

□

Remark 6.4.2. i. For an arbitrary monomial ideal J the set $\mathcal{F}(J)$ can be infinite. For example, if $J = (x) \lhd \mathbf{k}[x, y]$, $x < y$, then $\mathcal{F}(J) = \{xy^n \mid n \in \mathbb{N}\}$.

ii. Not all the complete systems turn out to be of the form of a star set.
For example, the complete system $M = \{x^h y,\, h \geq 1\} \subseteq \mathbf{k}[x, y]$ of Example 6.3.12 is not the star set of the ideal $J := (M)$.
Indeed, $\mathsf{N}(J) = \{x^m,\, m \geq 0\} \cup \{y^l,\, l > 0\}$ and all the terms of the form xy^k, $k > 1$, do not belong to M, even if $\frac{xy^k}{\min(xy^k)} = y^k \in \mathsf{N}(M)$.
Moreover, for $h > 1$, $\frac{x^h y}{x} = x^{h-1}y \in M$, so $x^h y \notin \mathcal{F}(J)$.

Better results hold if the monomial ideal J satisfies one of the following conditions, weaker then the strongly stable property (see section 6.6).

Definition 6.4.3. A monomial ideal J is called *stable* if it holds

$$\tau \in J,\ x_j > \min(\tau) \Longrightarrow \frac{x_j\tau}{\min(\tau)} \in J$$

A monomial ideal J is called *quasi stable* if it holds

$$\tau \in J,\ x_j > \min(\tau) \Longrightarrow \exists t \geq 0 :\ \frac{x_j^t\tau}{\min(\tau)} \in J.$$

We will show that this notion of quasi stable ideal coincides with the one given in [100], by proving that J actually has a Pommaret basis.

Remark 6.4.4.
- Obviously, a stably complete system M is also stable, and a stable set is also quasi stable.
- In order to verify whether the conditions above are satisfied for a given ideal J it is sufficient to check the terms in the basis $\mathsf{G}(J)$.

Proposition 6.4.5. Let J be a monomial ideal. Then TFAE:

i) J is stable

ii) $\mathcal{F}(J) = \mathsf{G}(J)$

Proof. *i)* $\Rightarrow$ *ii)* The inclusion $\mathsf{G}(J) \subseteq \mathcal{F}(J)$ is true for every monomial ideal by definition of star set. We prove now that $\gamma \notin \mathcal{F}(J)$ for every term $\gamma \in J \setminus \mathsf{G}(J)$.
By hypothesis, $\exists \tau \in \mathsf{G}(J)$, such that $\gamma = \tau\sigma$ and $\sigma \neq 1$.
Let $x_k := \min(\gamma)$. If $x_k|\sigma$, then $\frac{\gamma}{\min(\gamma)} = \tau\frac{\sigma}{x_k} \in J$, so that $\gamma \notin \mathcal{F}(J)$.

If, on the other hand, $x_k \nmid \sigma$ and x_j is any variable dividing σ, then $x_j > x_k$ and $x_k = \min(\tau)$. By the stability of J we have $\frac{x_j\tau}{x_k} \in J$, hence $\frac{\gamma}{x_k} = \frac{\tau\sigma}{x_j}\frac{x_j}{x_k} \in J$, hence again $\gamma \notin \mathcal{F}(J)$.

ii) $\Rightarrow$ *i)* If *ii)* holds, then $\mathsf{G}(J)$ is the only stably complete system generating J. By remark 6.4.4, we can check the stability on the terms $x^\alpha \in \mathsf{G}(J)$. Let $x_j > x_k := min(x^\alpha)$. By hypothesis there exists $x^\beta \in \mathsf{G}(J)$ such that $x_j x^\alpha \in \mathrm{off}_{\mathsf{G}(J)}(x^\beta)$, and, since $x^\alpha \in \mathsf{G}(J)$, of course $x^\alpha x_j \notin \mathsf{G}(J)$. Hence $x^\beta | \frac{x_j x^\alpha}{x_k}$ and so $\frac{x^\alpha x_j}{x_k} \in J$. □

Proposition 6.4.6. Let J be a monomial ideal. Then TFAE:

i) J is quasi stable

ii) $|\mathcal{F}(J)| < \infty$

iii) $\mathcal{F}(J) = \mathcal{H}(J)$ is the Pommaret basis of J.

Proof. *i)* $\Rightarrow$ *ii)* Let a be the maximum of the degrees of elements in $\mathsf{G}(J)$ and let t be such that $\frac{x_j^t x^\alpha}{\min(x^\alpha)} \in J$ for every $x^\alpha \in \mathsf{G}(J)$ and $x_j > min(x^\alpha)$. We prove that $\mathcal{F}(J)$ is contained in $\mathcal{P}_{<d}$ where $d := a + tn$. Let $x^\alpha x^\eta \in J_{\geq d}$ with $x^\alpha \in \mathsf{G}(J)$ and x_k be $\min(x^\alpha x^\eta)$. If $x_k|x^\eta$, then obviously $\frac{x^\alpha x^\eta}{x_k} = x^\alpha \frac{x^\eta}{x_k} \in J$, so $x^\alpha x^\eta \notin \mathcal{F}(J)$. If, on the other hand, $x_k \not| x^\eta$, then $x_k = \min(x^\alpha)$. Moreover, every variable dividing x^η is higher than x_k and at least one of them , let us call it x_j, appears in x^η with exponent $\geq t$, as $\deg(x^\eta) \geq nt$. Then $\frac{x_j^t x^\alpha}{x_k} \in J$, hence $\frac{x^\alpha x^\eta}{x_k} = \frac{x_j^t x^\alpha}{x_k} \cdot \frac{x^\eta}{x_j^t} \in J$ and $x^\alpha x^\eta \notin \mathcal{F}(J)$.

ii) $\Rightarrow$ *iii)* By *ii)* $\mathcal{F}(J)$ is finite, and by 6.4.1 is stably complete, so it is clearly the Pommaret basis of J.

iii) $\Rightarrow$ *i)* By remark 6.4.4, we check the quasi stability on the terms $x^\alpha \in \mathsf{G}(J)$. Let $x_j > x_k := min(x^\alpha)$. By the hypothesis on the finiteness of $\mathcal{F}(J)$, there exists $m \gg 0$ such that $x^\alpha x_j^m \notin \mathcal{F}(J)$. Moreover, being $\mathcal{F}(J)$ a stably complete system, there exists $x^\beta \in \mathcal{F}(J)$ such that $x_j^m x^\alpha \in \mathrm{off}_{\mathcal{F}(J)}(x^\beta)$ and $x^\beta | \frac{x_j^m x^\alpha}{x_k}$. Therefore, $\frac{x_j^m x^\alpha}{x_k} \in J$, namely J is quasi stable. □

Example 6.4.7. In $\mathbf{k}[x, y, z]$ with $x < y < z$:

- considered $J = (z, y^2)$, we get $M = \mathcal{F}(J) = \mathsf{G}(J) = \{z, y^2\}$, since J is stable;
- taken the ideal $J' = (z^2, y)$, we get $M = \mathcal{F}(J) = \{z^2, yz, y\} \supset \mathsf{G}(J)$.
  In fact, J is quasi stable, but it is not stable;
- given $J = (y)$, the star set is $M = \mathcal{F}(J) = \{z^k y \mid k \geq 0\}$, and $|\mathcal{F}(J)|$ is infinite, since J is not stable.

Remark 6.4.8. By remark 6.4.6, each zerodimensional ideal is quasi stable, since its star set is finite, as one can see by drawing the Bar Code of the corresponding Groebner escalier (see section 6.8 for more details).
Moreover, for non zerodimensional ideals, we can simply decide about their quasi stability by their (infinite) Bar Codes.
Indeed, by proposition 5.4.9, we only have to draw the corresponding infinite Bar Code and to check whether there is a finite term, under which lies at least one $\rightarrow$. If it is the case , the ideal is not quasi stable, since the star set is infinite. If not, the ideal is quasi stable.

6.5 M-marked sets and reduction process.

In this section, we generalize the notions of J-marked polynomial, J-marked basis and J-marked family given in [8, 27] for J strongly stable.
In those papers, the involved polynomials are marked on the monomial basis of the given monomial ideal J. Here, we give the analogous definitions for any monomial ideal, provided that the involved polynomials are marked on a complete generating system in the sense of definition 6.3.9.
After determining the setting, we extend to it the reduction process of the quoted papers.
At the end, we will see that such a generalized procedure does not need to be noetherian for every complete system of terms. We will need to add some hypotheses on the given complete system in order to overcome this problem.
We point out that, as in [8, 27], we do not introduce any term-ordering and this represents an important difference w.r.t. Janet's papers.
Moreover, we consider polynomials with coefficients in a ring, not necessarily in a field.

Definition 6.5.1. Let M be a complete system of terms and J be the ideal it generates.

- An *M-marked set* is a finite set $\mathcal{G}$ of homogeneous (monic) marked polynomials $f_\alpha = x^\alpha - \sum c_{\alpha\gamma} x^\gamma$, with $\mathrm{Ht}(f_\alpha) = x^\alpha \in M$ and $\mathrm{Supp}(f_\alpha - x^\alpha) \subset \mathsf{N}(J)$, so that $|\mathrm{Supp}(f) \cap J| = 1$.

- An *M-marked basis* $\mathcal{G}$ is an M-marked set such that $\mathsf{N}(J)$ is a basis of $\mathcal{Q}/(\mathcal{G})$ as A-module, i.e. $\mathcal{Q} = (\mathcal{G}) \oplus \langle \mathsf{N}(J) \rangle$ as an A-module.
- The *M-marked family* $\mathcal{M}f(M)$ is the set of all homogeneous ideals I that are generated by an M-marked basis.

Remark 6.5.2. Observe that the above definition of marked family $\mathcal{M}f(M)$ is consistent with that given in the Introduction of $\mathcal{M}f(J)$ for a monomial ideal J. Indeed, if $I \in \mathcal{M}f(M)$, then $I \in \mathcal{M}f(J)$ with $J = (M)$. On the other hand, for every given J there are complete systems M that generate it, for instance $M = \mathcal{F}(J)$ and $\mathcal{M}f(J) = \mathcal{M}f(M)$. In fact, if $I \in \mathcal{M}f(J)$, every polynomial h can be uniquely written as a sum $f + g$ with $f \in I$ and $g \in \langle \mathsf{N}(J) \rangle$; especially for every $x^\alpha \in M$, we have

$$x^\alpha = f_\alpha + g_\alpha,\ f_\alpha \in I \text{ and } g_\alpha \in \langle \mathsf{N}(J) \rangle. \tag{6.1}$$

Then I contains the M-marked basis

$$\mathcal{G} = \{ f_\alpha = x^\alpha - g_\alpha\ ,\ x^\alpha \in M \}.$$

Furthermore $\mathcal{G}$ is an M-marked basis since $(\mathcal{G}) \subseteq I$ and $\mathcal{Q} = (\mathcal{G}) + \langle \mathsf{N}(J) \rangle = I \oplus \langle \mathsf{N}(J) \rangle$.

The only difference between the two notations $\mathcal{M}f(J)$ and $\mathcal{M}f(M)$ with M a complete system generating J, is that using the second one we present every ideal of the family by means of a special set of generators depending on M. Note that, by the definition itself of $\mathcal{M}f(J)$, we can assert that for every ideal $I \in \mathcal{M}f(J)$ the M-marked basis generating it is *unique*.

We define now a reduction procedure for terms and polynomials, with respect to a homogeneous set $\mathcal{G}$ of polynomials, marked on a complete system of terms M.
The usual reduction process with respect to $\mathcal{G}$ consists of substituting each term $x^\alpha x^\eta$, multiple of an head term $x^\alpha = \mathrm{Ht}(f_\alpha)$, with the polynomial $(x^\alpha - f_\alpha)x^\eta = g_\alpha x^\eta$.
We add an extra condition to the standard procedure, namely that this substitution can be performed only in the case $x^\alpha x^\eta = x^\alpha *_M x^\eta$.

Definition 6.5.3. Let M be a complete system and $\mathcal{G}$ an M-marked set. We will denote by $\xrightarrow{\mathcal{G}}$ the transitive closure of the relation $h \xrightarrow{\mathcal{G}} h - cf_\alpha x^\eta$, where $x^\alpha x^\eta = x^\alpha *_M x^\eta$ is a term that appears in h with a non-zero coefficient c. We will say that $\xrightarrow{\mathcal{G}}$ is noetherian if the length r of any sequence $h = h_0 \xrightarrow{\mathcal{G}} h_1 \xrightarrow{\mathcal{G}} \dots \xrightarrow{\mathcal{G}} h_r$ is bounded by an integer number $m = m(h)$. This is equivalent to say that if we continue rewriting terms in this way we always obtain, after a finite number of reductions, a polynomial whose support is contained in $\mathsf{N}(J)$.

We will write $h \xrightarrow{\mathcal{G}}_* g$ if $h \xrightarrow{\mathcal{G}} g$ and $\mathrm{Supp}(g) \subset \mathsf{N}(J)$.

In general, the relation $\xrightarrow{\mathcal{G}}$ is not noetherian, namely there are sequences of reduction of infinite length.

Example 6.5.4. Let $M := \{xz, yz, y^2\}$ a set of terms in $\mathbf{k}[x, y, z]$ with $x < y < z$. We find the following sets of multiplicative variables:

- $mult_M(xz) = \{x, z\}$
- $mult_M(y^2) = \{x, y\}$
- $mult_M(yz) = \{x, y, z\}$

and check that M is complete.
Let $\mathcal{G}$ the M-marked set $\{f_{xz} = xz - xy,\ f_{yz} = yz - z^2,\ f_{y^2} = y^2\}$.
Then we have the infinite sequence of reductions:

$$xz^2 = xz *_M z \xrightarrow{G} xz^2 - f_{xz}z = xyz = yz *_M x \xrightarrow{\mathcal{G}} xyz - f_{yz}x = xz^2 ...$$

However, the reduction $\xrightarrow{\mathcal{G}}$ is always noetherian if $\mathcal{G}$ is marked on a stably complete system. In order to prove this fact we will use the following special subset of the ideal $(\mathcal{G})$.

Definition 6.5.5. Let $\mathcal{G}$ be an M-marked set on a complete system of terms M and let $J := (M)$. For each degree s, we will denote by $\mathcal{G}^{(s)}$ the set of homogeneous polynomial

$$\mathcal{G}^{(s)} := \{f_\alpha x^\eta \mid x^\alpha *_M x^\eta \in (M)_s\}$$

marked on the terms of J_s in the natural way $\mathrm{Ht}(f_\alpha x^\eta) = x^\alpha x^\eta$.

Remark 6.5.6. Observe that if $\mathcal{G}$ is a M-marked set on a stably complete system of terms M, for every homogeneous polynomial g of degree s, $g \xrightarrow{\mathcal{G}} h$ implies that $g - h = \sum_{i=1}^m c_i f_{\alpha_i} x^{\eta_i} \in \langle \mathcal{G}^{(s)} \rangle$.

It is worth noticing as a direct consequence of Lemma 6.3.23 that if $f_\alpha x^\eta \in \mathcal{G}$, then every term in $\mathrm{Supp}(x^\alpha x^\eta - f_\alpha x^\eta)$ either belongs to $\mathsf{N}((M))$ or is of the type $x^{\alpha'} *_M x^{\eta'}$ with $x^{\eta'} <_{Lex} x^\eta$.

Lemma 6.5.7. Let $\mathcal{G}$ be a M-marked set on the stably complete system of terms $M = \mathcal{F}(J)$.

1. Every term in $\mathrm{Supp}(x^\beta x^\epsilon - f_\beta x^\epsilon)$ either belongs to $\mathsf{N}((M))$ or is of the type $x^\alpha *_M x^\eta$ with $x^\eta <_{Lex} x^\epsilon$.
2. If $f_\beta \in \mathcal{F}(J)$, then all the polynomials $f_{\alpha_i} x^{\eta_i} \in \mathcal{G}^{(s)}$ used in the reduction of $x^\beta x^\epsilon$ (except $f_\beta x^\epsilon$ if it belongs to $\mathcal{G}^{(s)}$) are such that $x^\epsilon >_{Lex} x^{\eta_i}$.

3. If $g = \sum_{i=1}^{m} c_i f_{\alpha_i} x^{\eta_i}$, with $c_i \in k - \{0\}$ and $f_{\alpha_i} x^{\eta_i} \in \mathcal{G}^{(s)}$ are pairwise different, then $g \neq 0$ and its support contains some term of the ideal J.

Proof. (1) is a direct consequence of Lemma 6.3.23.

(2) Assume that the statement holds for every term $x^{\beta'} x^{\epsilon'}$, with $x^{\epsilon'} <_{Lex} x^{\epsilon}$. At a first step of reduction of $x^{\beta} x^{\epsilon}$ we use the polynomial $f_\alpha x^\eta$ where $x^\beta x^\epsilon = x^\alpha *_M x^\eta$, so that $x^\eta \leq_{Lex} x^\epsilon$; moreover every term in the support of the obtained polynomial either belongs to $\mathsf{N}((M))$ or is of the type $x^{\alpha'} *_M x^{\eta'}$ with $x^{\eta'} <_{Lex} x^\eta$ (Remark 6.5.6). Then we conclude since we assumed the property holds for all those terms.

(3) We assume that the summands in g are ordered so that $x^{\eta_1} \geq_{Lex} x^{\eta_i}$ for every $i = 1, \ldots, m$ and show that $x^{\eta_1 + \alpha_1}$ belongs to the support of g.

The term $x^{\alpha_1 + \eta_1}$ cannot appear as the head of $f_{\alpha_i} x^{\eta_i}$ for some $i \neq 1$ because the star decomposition of a term is unique. Moreover it cannot appear in $f_{\alpha_i} x^{\eta_i} - x^{\alpha_i + \eta_i}$ since $x^{\alpha_1 + \eta_1} = x^\beta x^{\eta_i}$, with $x^\beta \in \mathsf{N}(J)$ would imply $x^{\eta_i} >_{Lex} x^{\eta_1}$ (see Lemma 6.3.23), against the assumption. □

Theorem 6.5.8. Let $\mathcal{G}$ be an M-marked set on a stably complete system of terms M and let J be the ideal generated by M.

Then the reduction process $\xrightarrow{\mathcal{G}}$ is noetherian and, for every integer s, $\mathcal{Q}_s = \langle \mathcal{G}^{(s)} \rangle \oplus \langle \mathsf{N}(J)_s \rangle$. Indeed, for every $h \in \mathcal{Q}_s$

$$h = f + g \text{ with } f \in \langle \mathcal{G}^{(s)} \rangle \text{ and } g \in \langle \mathsf{N}(J)_s \rangle \iff h \xrightarrow{\mathcal{G}}_* g \text{ and } f = h - g$$

Proof. Let $\mathcal{G} = \{f_\alpha \mid x^\alpha \in M\}$.
We observe that we have $\langle \mathcal{G}^{(s)} \rangle \cap \langle \mathsf{N}(J)_s \rangle = \{0\}$ by Lemma 6.5.7.

In order to prove that the module $\langle \mathcal{G}^{(s)} \rangle + \langle \mathsf{N}(J)_s \rangle$ coincides with $\mathcal{P}_s$ it is sufficient to show that it contains all the terms in $J_s \setminus M$, being obvious for those in M, for which $x^\alpha = f_\alpha + g_\alpha$ (see 6.1).

Let τ be a term in J_s.
If $\tau = x^\alpha *_M x^\eta$, we may assume of having already proved the statement for all the terms $\tau' = x^{\alpha'} *_M x^{\eta'}$ with $x^{\eta'} <_{Lex} x^\eta$.

We have $x^\alpha x^\eta = f_\alpha x^\eta + (x^\alpha - f_\alpha) x^\eta$ where $\mathrm{Supp}(x^\alpha - f_\alpha) \subset \mathsf{N}(J)$. If x^β is any term in this support, then either $x^{\beta+\eta} \in \mathsf{N}(J)$ or $x^{\beta+\eta} = x^{\alpha'} *_M x^{\eta'}$ with $x^{\eta'} <_{Lex} x^\eta$ by Lemma 6.3.23. This allows us to conclude $\mathcal{Q}_s = \langle \mathcal{G}^{(s)} \rangle + \langle \mathsf{N}(J)_s \rangle$.

Finally, in order to prove that $\xrightarrow{\mathcal{G}}$ is noetherian it is sufficient to observe that every reduction step substitutes a term of J of the type $x^\alpha *_M x^\eta$ with $x^\alpha x^\eta - f_\alpha x^\eta$. Indeed, by remark 6.5.6, each $\tau \in \mathrm{Supp}(x^\alpha x^\eta - f_\alpha x^\eta) \setminus \mathsf{N}((M))$ has the form $x^{\alpha'} *_M x^{\eta'}$, $x^{\eta'} <_{Lex} x^\eta$ and this permits to conclude by induction. □

As a straightforward consequence of the previous result, we obtain the following

Corollary 6.5.9. If M is a stably complete system and $\mathcal{G}$ is an M-marked set, the following are equivalent:

- $\mathcal{G}$ is an M-marked basis
- for every s: $\langle \mathcal{G}^{(s)} \rangle = (\mathcal{G})_s$
- for every $h \in (\mathcal{G})$: $h \xrightarrow{\mathcal{G}}_* 0$
- if $h - g \in (\mathcal{G})$ and $\mathrm{Supp}(g) \subset \mathsf{N}(J)$, then $h \xrightarrow{\mathcal{G}}_* g$.

Remark 6.5.10. We point out that if $\mathcal{G}$ is a M-marked set, but not a M-marked basis, then there are polynomials in the ideal $(\mathcal{G})$ whose support is contained in $\mathsf{N}((M))$. Hence, we do not have a "normal form" of a polynomial h modulo $(\mathcal{G})$, since, in general, there are several polynomials g' such that $\mathrm{Supp}(g') \subset \mathsf{N}(J)$ and $h - g' \in (\mathcal{G})$. However, the reduction process $h \xrightarrow{\mathcal{G}}_* g$ with respect to a $\mathcal{F}(J)$-marked set $\mathcal{G}$ gives a unique reduced polynomial g for every polynomial h.

Using the reduction process, we can now answer Problem 6.1.2 and characterize the ideals I that belong to the marked family $\mathcal{M}f(J)$.

Theorem 6.5.11. Let $\mathcal{G}$ be a $\mathcal{F}(J)$-marked set. Then:

$$(\mathcal{G}) \in \mathcal{M}f(J) \Longleftrightarrow \forall f_\beta \in \mathcal{G},\ \forall x_i > \min(x^\beta) :\ f_\beta x_i \xrightarrow{\mathcal{G}}_* 0$$

Proof. Since "$\Rightarrow$" is a straightforward consequence of Corollary 6.5.9, we only prove "$\Leftarrow$". More precisely, we prove that $(\mathcal{G})_m = (\mathcal{G}_{(m)})$, showing that if $f_\beta \in \mathcal{G}$ and $\deg(x^{\beta+\epsilon}) = m$, then $f_\beta x^\epsilon$ is either an element of $\mathcal{G}_{(m)}$ itself or a linear combination of polynomials in $\mathcal{G}_{(m)}$. If this were not true, we can choose an element $f_\beta x^\epsilon \notin \langle \mathcal{G}_{(m)} \rangle$ with x^ϵ minimal with respect to $<_{Lex}$. As $f_\beta x^\epsilon \notin \mathcal{G}_{(m)}$, at least one variable x_i appearing in x^ϵ with nonzero exponent is non-multiplicative for x^β. Let $x^\epsilon = x_i x^{\epsilon'}$. By hypothesis $f_\beta x_i \xrightarrow{\mathcal{G}}_* 0$, so that $f_\beta x_i$ is a linear combination $\sum c_i f_{\alpha_i} x^{\eta_i}$ of polynomials in $\mathcal{G}_{(|\beta|+1)}$. By Lemma 6.5.7 we have $x^{\eta_i} <_{Lex} x_i$. Now $f_\beta x^\epsilon = (f_\beta x_i) x^{\epsilon'} = (\sum c_i f_{\alpha_i} x^{\eta_i}) x^{\epsilon'} = \sum c_i f_{\alpha_i} x^{\eta_i + \epsilon'}$, where $x^{\eta_i+\epsilon'} <_{Lex} x_i x^{\epsilon'} = x^\epsilon$. Now we get a contradiction, since $f_{\alpha_i} x^{\eta_i+\epsilon'} \in \langle \mathcal{G}_{(m)} \rangle$ by the minimality of x^ϵ. □

Example 6.5.12. Let J be the monomial ideal (x^3, xy, y^3) in $\mathbf{k}[x, y]$ with $x < y$. Its star set is $\mathcal{F}(J) = \{x^3, xy, xy^2, y^3\}$. Using the criterion given in Theorem 6.5.11, we can easily check that the $\mathcal{F}(J)$-marked set $\mathcal{G} := \{f_1 := \mathbf{x^3},\ f_2 := \mathbf{xy} - x^2 - y^2,\ f_3 := \mathbf{xy^2},\ f_4 = \mathbf{y^3}\}$ (in bold the head terms) is a $\mathcal{F}(J)$-market basis:

- $yf_1 = xf_1 + x^2f_2 + xf_3 \xrightarrow{\mathcal{G}}_* 0$,
- $yf_2 = f_1 - xf_2 - f_4 \xrightarrow{\mathcal{G}}_* 0$
- $yf_3 = xf_4 \xrightarrow{\mathcal{G}}_* 0$.

This is a simple example of a marked basis which is not a Groebner basis. In fact, it is obvious that $\mathrm{Ht}(f_2) = xy$ cannot be the leading term of f_2 with respect to any term-ordering and, more generally, that J cannot be the initial ideal of the ideal $(\mathcal{G})$, even though $(\mathcal{G}) \oplus \mathsf{N}(J) = \mathbf{k}[x, y]$.

A wider family of ideals of this type are presented in [27, Example 3.18 and Appendix].

Remark 6.5.13. Let $x^\beta x_i = Ht(f_\beta x_i)$, if $x^\beta x_i = x^\alpha x^\eta \in \mathrm{off}(x^\alpha)$ then the first step of reduction of the polynomial $f_\beta x_i$ is actually $f_\beta x_i \xrightarrow{\mathcal{G}} S(f_\beta, f_\alpha) := \frac{lcm(x^\beta, x^\alpha)}{x^\beta} f_\beta - \frac{lcm(x^\beta, x^\alpha)}{x^\alpha} f_\alpha = f_\beta x_i - f_\alpha x^\eta$, namely the *S-polynomial* of f_α, f_β. Therefore we could reformulate the criterion given by Theorem 6.5.11 as follows:

$$(\mathcal{G}) \in \mathcal{M}f(J) \iff \forall f_\alpha, f_\beta \in \mathcal{G} : \; S(f_\alpha, f_\beta) \xrightarrow{\mathcal{G}}_* 0.$$

However Theorem 6.5.11 shows that it is sufficient to check a special subset of the S-polynomials that corresponds to the basis for the first syzygies of the terms in $\mathcal{F}(J)$. If J is quasi stable, this basis is the one considered in [99]. It is obvious that the maximal degree of these special S-polynomials cannot exceed $1 + \max\{\deg(x^\alpha) \mid x^\alpha \in \mathcal{F}(J)\}$. Indeed, if J is quasi stable, $reg(J) = \max\{deg(\tau), \tau \in \mathcal{F}(J)\}$ as proved in [48, 54, 100].

Remark 6.5.14. If J is a quasi stable monomial ideal and $\mathcal{G}$ is an $\mathcal{F}(J)$-marked set, then there are only a finite number of reduction to perform in order to decide if a $\mathcal{F}(J)$-marked set $\mathcal{G}$ is a basis. We will use this algorithm in order to endow the marked family $\mathcal{M}f(J)$ of a structure of affine scheme

If the considered monomial ideal is not quasi stable, then the (unique) stably complete generating set is *infinite*. Actually this does not necessarily exclude we can exploit it even from a computational point of view.

6.6 Marked families, schemes and functors

In this section we follow [8, 27] and show how it is possible to associate a scheme to each marked family $\mathcal{M}f(J)$. Due to the naturality of this construction, we can mimic the one of [64], and define marked families as functors.

Our results are very similar, but more general, than those of [8, 27, 64]; in fact in those papers the ideal J is assumed to be strongly stable.

Obviously, a strongly stable ideal is also stable, so that $\mathcal{F}(J) = \mathsf{G}(J)$. If J is strongly stable, the notions of $\mathsf{G}(J)$-marked sets, $\mathsf{G}(J)$-marked bases and $\mathsf{G}(J)$-marked family introduced in the previous sections exactly correspond to those of J-marked sets, J-marked bases, J-marked family considered in [8, 27] and the reduction procedure $\xrightarrow{\mathcal{G}}$ with respect to a $\mathsf{G}(J)$-marked set $\mathcal{G}$ introduced in definition 6.5.3 coincides with the one used in those papers.

Moreover, for such an ideal J, the scheme structure that we will define is the same obtained in [8, 27] and used in [10, 64] for a local study of Hilbert schemes. Indeed, for every monomial ideal J, if $I \in \mathcal{M}f(J)$, then the ideals I and J share the same Hilbert polynomial (and also the same Hilbert function), so that they correspond to points in the same Hilbert scheme.

The scheme we associate to $\mathcal{M}f(J)$ only depends on the monomial ideal J, but the way we use in order to define it needs a set of generators M complete, finite and such that for every M-marked set $\mathcal{G}$ the reduction procedure $\xrightarrow{\mathcal{G}}$ is noetherian.

Then, in the following J will be a quasi stable monomial ideal and M will be its finite star-set $\mathcal{F}(J)$, (according to Seiler's notation, it is the Pommaret basis $\mathcal{H}(J)$).

Let $\{x^{\alpha_1}, ..., x^{\alpha_s}\}$ be the terms in M and consider the polynomial ring $B := A[C]$, where C is a compact notation for the set of variables $C_{i,\beta}$ $i = 1, \ldots, s$ and $x^\beta \in \mathsf{N}(J)_{deg(\alpha_i)}$. We also define the M-marked set in $B[x_1, ..., x_n]$

$$\mathcal{G} := \{f_{\alpha_i} := x^{\alpha_i} + \sum C_{i,\beta} x^\beta \mid x^\beta \in \mathsf{N}(J)_{|\alpha_i|}, \mathrm{Ht}(f_{\alpha_i}) = x^{\alpha_i}\}.$$

Clearly, every M-marked set can be obtained specializing $\mathcal{G}$, namely as $\phi(\mathcal{G})$ for a suitable morphism of A-algebras $\phi : A[C] \to A$. Moreover, by the uniqueness of the M-marked basis generating each ideal in $\mathcal{M}f(J)$, we can assert that for every ideal $I \in \mathcal{M}f(J)$ there exists a unique specialization ϕ such that $(\phi(\mathcal{G})) = I$.

We use Theorem 6.5.11 in order to construct a set of polynomials $\mathcal{R}$ that will define the scheme we associate to M. If g is a polynomial in $B[x_1, ..., x_n]$, we denote by $\mathrm{coeff}_x(g)$ the set of coefficients of g with respect to the only set of variables $x_1, \ldots, x_n$; hence $\mathrm{coeff}_x(g) \subset B = A[C]$ is a set of polynomials in the variables C. For every $x^{\alpha_i} \in M$ and $x_j > \min(x^{\alpha_i})$, let $g_{\alpha_i,j} \in B[x_1, \ldots, x_n]$ be such that $f_{\alpha_i} x_j \xrightarrow{\mathcal{G}}_* g_{\alpha_i,j}$.

Definition 6.6.1. Let M be a stably complete system in $\mathcal{T}$, A be any ring, and $\mathcal{R}$ be the union of $\mathrm{coeff}_x(g_{\alpha_i,j})$ for every $x^{\alpha_i} \in M$ and $x_j > \min(x^{\alpha_i})$.

We will call M*-marked scheme* over the ring A, and denote by $\mathbf{Mf}_M(A)$ the affine scheme $\mathrm{Spec}(A[C]/(\mathcal{R}))$.

Remark 6.6.2. Every M-marked set in $A[x_1, \ldots, x_n]$ is a M-marked basis if and only if the coefficients of the terms in the tails satisfy the conditions given by $\mathcal{R}$.

In particular, if $A = k$ is an algebraically closed field, then the closed points of $\mathbf{Mf}_M(A)$ correspond to the ideals in the marked family $\mathcal{M}f(J)$ where J is the ideal in $\mathbf{k}[x_1, \ldots, x_n]$ generated by M.

Remark 6.6.3. The above construction of $\mathcal{R}$ is in fact *independent* from the fixed commutative ring A, in the sense that it is preserved by extension of scalars. We can first choose $\mathbb{Z}$ as the coefficient ring and then apply the standard map $\mathbb{Z} \to A$.

More formally, for every stably complete set of terms M we can define a functor between the category of $\mathbb{Z}$-algebras to the category of sets

$$\underline{\mathbf{Mf}}_M : \underline{\mathbb{Z}\text{-Alg}} \to \underline{\mathrm{Set}}$$

that associates to any $\mathbb{Z}$-algebra A the set $\underline{\mathbf{Mf}}_M(A) := \mathcal{M}f(MA[x_1, \ldots, x_n])$ and to any morphism $\phi : A \to B$ the map

$$\begin{array}{rccc} \underline{\mathbf{Mf}}_J(\phi): & \underline{\mathbf{Mf}}_M(A) & \longrightarrow & \underline{\mathbf{Mf}}_M(B) \\ & \mathrm{I} & \longmapsto & I \otimes_A B. \end{array}$$

Moreover, again following [64], it is possible to prove that $\underline{\mathbf{Mf}}_M$ is a representable functor represented by the scheme $\mathbf{Mf}_M(\mathbb{Z}) = \mathrm{Spec}(\mathbb{Z}[C]/(\mathcal{R}))$.

6.7 Historical notes.

Through the trivial interpretation of derivatives

$$\frac{1}{\alpha_1! \cdots \alpha_n!} \frac{\partial^{\alpha_1+\alpha_2+\ldots+\alpha_n}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_n^{\alpha_n}},$$

in terms of the corresponding term $\tau = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n} \in \mathcal{T}$, Riquier [87, 88, 89] was able to algebraically transform the problem of solving differential partial equations in terms of ideal membership.

After introducing the concept (but not the notion) of S-polynomials he proved that if the normal form (in terms of Gauss-Buchberger reduction) of each S-polynomial among the elements of the basis $\mathcal{G}$ goes to zero then

- the given basis $\mathcal{G}$ generates the related ideal;
- the generic solution of the PDE can be given (and computed) as series in terms of initial conditions which can be described and formulated in terms of a Hironaka-Galligo-like decomposition [38, 51] (but more general) of the related *escalier* N;

if not all normal forms are 0, then, exactly as in Buchberger Algorithm, the non-zero normal forms are included in the basis and the procedure is repeated.

For instance, the system [89, pp.188-9]

$$\frac{\partial^3 u}{\partial y^3} = A(x,y,z), \quad \frac{\partial^2 u}{\partial x \partial z} = B(x,y,z), \quad \frac{\partial^3 u}{\partial x^2 \partial y} = C(x,y,z),$$

must satisfy the integrability conditions

$$\frac{\partial^2 A}{\partial x \partial z} = \frac{\partial^3 B}{\partial y^3}, \quad \frac{\partial^2 A}{\partial x^2} = \frac{\partial^2 C}{\partial y^2}, \quad \frac{\partial^2 B}{\partial x \partial y} = \frac{\partial C}{\partial z};$$

in which case the initial conditions have the shape

$$\left\{ \begin{array}{rcll} \left.\begin{array}{rcl} u & = & \phi_0(z) \\ \frac{\partial u}{\partial y} & = & \phi_1(z) \\ \frac{\partial^2 u}{\partial y^2} & = & \phi_2(z) \end{array}\right\} & & x - x_0 = y - y_0 = 0, \\ \\ \left.\begin{array}{rcl} \frac{\partial u}{\partial x} & = & \alpha_0 \\ \frac{\partial^2 u}{\partial x \partial y} & = & \alpha_1 \\ \frac{\partial^3 u}{\partial x \partial y^2} & = & \alpha_2 \end{array}\right\} & & x - x_0 = y - y_0 = z - z_0 = 0, \\ \\ \frac{\partial^2 u}{\partial x^2} = \psi(x) & & y - y_0 = z - z_0 = 0. \end{array} \right.$$

In his theory, Riquier was assuming that the set $\mathcal{T}$ of the terms was ordered by a term-ordering; he was mainly using [89, p.67] the deglex ordering induced by $x_1 > x_2 > \cdots > x_n$, but he gave a large class of term-orderings to which his theory was applicable; actually (but he never stated that) his characterization is the classical one of all term-orderings [34, 90]. He was however forced to restrict himself to degree-compatible term-orderings in order to be granted convergency.

In his gaussian reduction, Riquier, as Buchberger, considered as head term of each "marked" polynomial its maximal term.

In his considerations on generic initial ideal, Delassus [32], followed by Robinson [91] used (deg)-rev-lex induced by $x_1 < x_2 \cdots < x_n$ and the minimal term as head term of each “marked” polynomial.

In order to "harmonize" the two notations, Janet in [54, 57] applied deglex induced by $x_1 < x_2 < \cdots < x_n$ and chose the maximal term as head term, but expressed all terms as (!) $x_n^{\alpha_n} x_{n-1}^{\alpha_{n-1}} \dots x_1^{\alpha_1}$, while in [55] went back to use deglex induced by $x_1 > x_2 > \cdots > x_n$.

What is worst, in [56] Janet not only applied deglex induced by $x_1 < x_2, \cdots < x_n$ but presented all results within his notation; so, in his presentation of Delassus's result, the head term is again, *à la* Buchberger, the maximal one.

This is not helpful, as regards his reformulation of the previous results on generic initial ideals and stability; thus while, for Robinson [91, 92] and Gunther [46, 47] a generic initial ideal $\epsilon(I)$ satisfies

$$\mu \in \epsilon(I),\, x_h \mid \mu, i < h \implies x_i \frac{\mu}{x_h} \in \epsilon(I),$$

according [56] the formula is

$$\mu \in \epsilon(I), x_h \mid \mu, i > h \implies x_i \frac{\mu}{x_h} \in \epsilon(I).$$

Under the suggestion of Hadamard [84], Janet dedicated his doctorial thesis [54] to a reformulation of Riquier's results in terms of Hilbert's results [50].

In particular, given a finite set of monomials M, he associates to each term $\tau \in M$, as functions of its relation with the other elements of M, a set of variables which he labels *multiplicative* (Definition 6.3.1) and a subset of terms in (M) which he called his *class* and which we labeled as its *offspring* and considered M *complete* (Definition 6.3.9) when the disjoint offsprings of M cover (M).

He then gave [54, p.80] a *procédé régulier pour obtenir un système complet base d'un module donné* which *ne pourra se prolonger indéfiniment*; it simply consisted to enlarge M with the elements $xt \notin \cup_{\tau \in M} \mathrm{off}_M(\tau), t \in M$, x non-multiplicative for t.

Janet can now formulate [57, p.75] Riquier's procedure. One can assume to have a finite basis $\mathcal{G} \subset \mathcal{P}$; denoting $M = \{\mathsf{T}(f) : f \in \mathcal{G}\}$,

- we enlarge M in order to made it complete and at the same time
- we similarly enlarge $\mathcal{G}$, adding xg to $\mathcal{G}$ when we add $x\mathsf{T}(g) \notin \cup_{\tau \in M} \mathrm{off}_M(\tau)$;
- we then perform Riquier's test, which, for a complete systems, consists in computing the normal form of each element $xg, g \in \mathcal{G}$, x non-multiplicative for $\mathsf{T}(g)$.

Janet [54, p.112-3] further remarks (in connection with Hilbert's syzygy theory) that the reduction-to-zero of all such elements give a basis S of the syzygy module of $\mathcal{G}$. Actually he repeatedly applied the same procedure to S, thus computing a resolution of $\mathcal{G}$ and anticipating Schreyer's Algorithm [94].

Next, in 1924, Janet [55] moved his interest in extending the study to the homogeneous case, adapting his approach on one side to the solution of partial differential equation given by E. Cartan [14, 15, 16] via his characters and test and on the other side to the introduction by Delassus [32] of the concept of generic initial ideal and the precise description of it given by Robinson [91, 92] and Gunther [46, 47]; he thus discussed the notion of *système de forms (de même ordere) en involution*. The notion, as he explains, is independent from the variable chosen and allows to assign to the system a series of values $\sigma_i^{(p)}, 1 \leq i \leq n, p \in \mathbb{N}$ which [57, p.87] *sont évidemment invariables lorsq'on fait un changement linéaire et homogène des variables indépendantes* which, under the assumption of generality, allow to describe the structure of the *generic escalier* of the considered ideal.
The procedure, given a finite set $\mathcal{G}$ of forms, repeatedly produces *à la* Macaulay a linear basis $\mathcal{B}_p$ of $(\mathcal{G})_p$ by performing linear algebra on the set $\{x_i g : g \in \mathcal{B}_{p-1}, 1 \leq i \leq n\}$; termination is granted when the formula (6.2) below is satisfied.

Given a homogeneous ideal $I \subset \mathbf{k}[x_1, x_2, \ldots, x_n]$, where the variables are assumed to be generic, so that $\mathsf{N}(I)$ is stable, Janet defined [55, pp.30-2],[56, p.30],[57, pp.90-1],[84, p.93, p.99] multiplicative variables according 6.3.6, introduced values $\sigma_i^{(p)}(I)$ (or $\sigma_i^{(p)}$ for short when no confusion is possible) for every $1 \leq i \leq n$, and $p \in \mathbb{N}$, which can be described as

$$\sigma_i^{(p)} := \#\left\{\tau \in \mathsf{N}(I), \deg(\tau) = p, \min(\tau) = i\right\}$$

and, fixing a value p and denoting $\sigma_i := \sigma_i^{(p)}$, and $\sigma_i' := \sigma_i^{(p+1)}$ proved

Proposition 6.7.1 (Janet)**.** It holds,

1. $\sigma_1' + \sigma_2' + \ldots + \sigma_n' \leq \sigma_1 + 2\sigma_2 + \ldots + n\sigma_n$;
2. $\sum_{i=1}^n \sigma'_i = \sum_{i=1}^n i\sigma_i \implies \sigma'_j = \sum_{i=j}^n \sigma_i$ for each j.
3. $\sum_{i=1}^n \sigma'_i = \sum_{i=1}^n i\sigma_i \implies \sum_{i=1}^n \sigma_i^{(P+1)} = \sum_{i=1}^n i\sigma_i^{(P)}$ for each $P > p$.

He can then state

Definition 6.7.2 (Janet)**.** [57, pp.90-1] A finite set $E \subset \mathcal{P}$ of forms of degree at most p generating the ideal $I \subset P$, is said to be *involutive*[16] if, with the present notation, it satisfes the formula

$$\sum_{i=1}^n \sigma_i^{(p+1)} = \sum_{i=1}^n i\sigma_i^{(p)}. \tag{6.2}$$

□

[16] *en involution.*

Thus, once the iterated Macaualy-like procedure satisfies (6.2) at degree $\bar{p}$ then it successfully terminates and the *finite* bases produced by it is involutive; Janet is therefore able to present the ideal $\{\tau \in \mathsf{T}(I), \deg(\tau) \geq \bar{p}\}$ by explicitly producing[57] the decomposition

$$\{\tau \in \mathsf{T}(I), \deg(\tau) \geq \bar{p}\} = \sqcup_{\tau \in M} \mathrm{off}_M(\tau)$$

where M is the stably complete set $M = \{\tau \in \mathsf{T}(I), \deg(\tau) \geq \bar{p}\}$ and to express its Hilbert polynomial as

$${}^h H_I(t) = \sum_{h=1}^{n-1} \binom{t-p+h-1}{h-1} \sigma_h^{(p)}(I).$$

In our context, the characterization of $\sigma_i^{(p)}$ and definition 6.7.2 lead to the following

Proposition 6.7.3. With the previous notation, if J is a quasi stable monomial ideal, then

$$\sum_{i=1}^{n} \sigma_i^{(p+1)}(J) = \sum_{i=1}^{n} i\sigma_i^{(p)}(J).$$

The same equality holds if I is a homogeneous ideal generated by a J-marked basis $\mathcal{G}$ with J quasi stable.

Therefore $\mathcal{G}$ is an involutive basis.

Proof. For the first statement we observe that if $p \geq \overline{p}$ every term $\tau \in J_{p+1}$ can be written in a unique way as a product $\tau = \theta x_i$, with $\theta \in J_p$ and x_i a multiplicative variable for θ, i.e. $x_i \leq \min(\theta)$.

If I is the homogeneous ideal generated by a J-marked set $\mathcal{G}$, then for the corresponding $f_\tau \in \mathcal{G}_{(p+1)}$ we have $f_\tau = f_\theta x_i$ with f_θ in $\mathcal{G}_{(p)}$ and of course $x_i \leq \min(\theta)$.

If $\mathcal{G}$ is a J-marked basis, then we get the equality since $(\mathcal{G})_t = (\mathcal{G}_{(t)})$ for every t (Corollary 6.5.9). □

Note that for an ideal I generated by a J-marked set $\mathcal{G}$ which is not a marked basis, only the inequality $\sum_{i=1}^n \sigma_i^{(p+1)} \leq \sum_{i=1}^n i\sigma_i^{(p)}$ holds true, since $(\mathcal{G})_t \supseteq (\mathcal{G}_{(t)})$.

The iterated Macaualy-like procedure gives also a fine decomposition of $\mathsf{N}(I)_{\geq \bar{p}-1}$ as follows:

- Janet partitions the set $\mathsf{N}(I)_{\overline{p}-1}$ as $\mathsf{N}_{\overline{p}-1} = \sqcup_{i=0}^{n-1} N_i$ associating to
  - N_0 the monomials $\tau \in \mathsf{N}_{\overline{p}-1}(I)$ for which $x_1\tau \in \mathsf{T}(I)$;
  - while each of the σ_1 elements $\tau = \frac{v}{x_1} \in \mathsf{N}(I)_{\overline{p}-1} \setminus N_0, \mathrm{class}(v) = 1$[17], is inserted in N_i if it is one of the σ_i elements which can be expressed as $\tau = \frac{v_i}{x_i}, \mathrm{class}(v_i) = i$

[17] In this context, a term $\omega \in \mathcal{T}$ has $\mathrm{class}(\omega) = i$ if $\omega \in \mathcal{T}[i,n] \setminus \mathcal{T}[i+1,n]$.

but is not one of the σ_{i+1} elements which can be expressed as $\tau = \frac{v_{i+1}}{x_{i+1}}, \mathrm{class}(v_{i+1}) = i+1$.

- he then associates to each $\tau \in N_i$ a set $\mathrm{mult}(\tau) = \{x_j, 1 \leq j \leq i\}$ of multiplicative variables and a set $\mathrm{off}(\tau) := \{\tau\omega, \omega \in \mathcal{T}[1,i]\}$ as its offspring
- and states

$$\{\tau \in \mathsf{N}(I), \deg(\tau) \geq \bar{p} - 1\} = \sqcup_{i=0}^{n-1} \sqcup_{\tau \in N_i} \mathrm{off}(\tau).$$

Riquier's and Janet's results were introduced to the Computational Algebra commutative at the MEGA-90 Symposium in 1990 by a survey by Pommaret [85] of his theory and, two years later, through a paper by F. Schwarz [95] where he remarked:

> The concept of a Gröbner base and algorithmic methods for constructing it for a given system of multivariate polynomials has been established as an extremely important tool in commutative algebra. It seems to be less well known that similar ideas have been applied for investigating partial differential equations (pde's) around the turn of the century in the pioneering work of the French mathematicians Riquier and Janet. [...] their theory [...] is basically a critical-pair/completion procedure. All basic concepts like term-ordering, reductions and formation of critical pairs are already there.

This prompted V. Gerdt to suggest his coworkers Zharkov and Blinkov to investigate whether the results by Janet and Pommaret could be translated from pde's to polynomial rings in order to produce an effective alternative approach to Buchberger's Algorithm; the conclusion of this investigation [107, 108] was successful — the proposed algorithm was able to give a solution with a speed-up of 20 w.r.t. degrevlex Buchberger's algorithm on classical test-suites and caused sensation in the community.

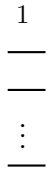
Unfortunately, among the two constructions proposed by Janet, they hitted the involutive one, which is *not* a Buchberger-like procedure and presented it as such, remarking that in general does not terminate and that the basis is not necessarily finite unless the ideal is 0-dimensional. What is worst, they attributed to Pommaret their mistakes, thus introducing in literature a "bad" fictional Pommaret division compared with the "good" Janet division (related to Janet completion [54] procedure).

An algorithm based on Janet's notion [54] of completeness is reported in [40, 41, 42]

Involutiveness is the argument of the *Habilitation* thesis (2002) of Seiler [97, 99, 100]; an improved version has recently appeared as [98]. Finiteness is a required condition for the notion of Pommaret bases [48].

6.8 An involutive Moeller Algorithm.

In this section we develop a version of Moeller algorithm which computes a lexicographical reduced involutive basis for a zero-dimensional radical ideal I, requiring only the finite set of distinct points $\mathbf{X} := V(I)$.
Consider a finite set of distinct points $\mathbf{X} = \{P_1, ..., P_S\}$. As explained while talking about the Bar-Code Axis of Evil algorithm, if $\mathbf{X} = \{P_1\}$, $P_1 = (a_{1,1}, ..., a_{1,n})$, the ideal $I = I(\mathbf{X})$ is the maximal ideal $I = (x_1 - a_{1,1}, ..., x_n - a_{1,n})$.
The initial ideal $J = In_<(I) = (x_1, ..., x_n)$ is quasi stable, being zerodimensional (6.4.8). As a matter of fact, given $\tau \in J$, $\exists x_h \mid \tau$, $1 \leq h \leq n$. Consider now $x_j > \min(\tau)$ and compute $\sigma = \frac{x_j \tau}{\min(\tau)}$. Clearly $x_j \mid \sigma$, so $\sigma \in J$ and J is definitively quasi stable.
We can also get the quasi stability of J using the Bar Code. In fact, the Groebner escalier associated to J is $\mathsf{N}(J) = \{1\}$ and the Bar Code is



The star set is then $\mathcal{F}(J) = \{x_1, ..., x_n\}$ and equals the monomial basis $\mathsf{G}(J)$, so J is stable by proposition 6.4.5.
Clearly, this implies that the reduced Groebner basis $\mathcal{G}_1 = \{x_1 - a_{1,1}, ..., x_n - a_{1,n}\}$ is also the reduced involutive basis. $\mathcal{J}_1$
We point out that the polynomials in $\mathcal{J}_1$ are ordered. More precisely, the first polynomial is the one whose leading term is $x_1 = \min(\mathsf{G}(J))$. The leading term of the second polynomial is $x_2 > x_1$ and so on. The last polynomial is $x_n - a_{1,n}$ and $x_n = \max(\mathsf{G}(J))$. We say, by abuse of notation, that the polynomials are *ordered with respect to lex*.
The triangular polynomial for $\{P_1\}$ is $q_1 = 1$.
We consider the data obtained for the singleton $\{P_1\}$ as the basis for our procedure in the case $|\mathbf{X}| = m > 1$.
In this setting, we consider the point $P_2 = (a_{2,1}, ..., a_{2,n})$. If, for some $j \in \{1, ..., n\}$, $a_{1,j} = a_{2,j}$, the polynomial $x_j - a_{1,j} \in \mathcal{J}_1$ computed before vanishes in P_2.
This implies that if $f \in \mathcal{J}_1$ is the minimal polynomial with respect to lex not vanishing in P_2 and $x_j = T(f)$, $j \leq n$, then P_2 shares the first $j-1$ coordinates with P_1: $a_{1,1} = a_{2,1}, ..., a_{j-1,1} = a_{j-1,2}$.
As seen while talking about the original Moeller algorithm, $T(f)$ is the term corresponding to P_2 in the Groebner escalier $\mathsf{N}(\{P_1, P_2\})$.

Since $T(f) \in \mathsf{N}(\{P_1, P_2\})$, it cannot belong to the minimal basis anymore, so we remove f from the Groebner basis.
More precisely f vanishes in P_1, while $f(P_2) \neq 0$. We construct then $q_2 = \frac{1}{f(P_2)} f$ which is the second triangular polynomial.
All the polynomials in $\mathcal{J}_1$ whose leading term is smaller than $T(f)$ vanish in P_2, and so they automatically belong to $\mathcal{J}_2$, but we cannot assert the same for the polynomials $g \in \mathcal{J}_1$ with $\mathsf{T}(g) > \mathsf{T}(f)$ so we need to interpolate them in P_2. The polynomials obtained this way belong to $\mathcal{J}_2$.
By proposition 5.7.1, now, we have to insert in $\mathcal{J}_2$ the polynomials $\tau - Can(\tau, I(\{P_1, P_2\}))$, for $\tau \in \{x_j T(f),\, x_j \leq \min(\mathsf{T}(f))\}$.
In order to compute them, we only have to perform the interpolating procedure GaussRed from the original Moeller algorithm on these terms. Once this step is completed, we get $\mathcal{J}_2$.
Suppose now to have computed $\mathcal{J}_{i-1}$ and let us explain the steps to perform for the point P_i.

- Find $\tau = \min\{\mathsf{T}(f) \,|\, f \in \mathcal{J}_{i-1} \text{ and } f(P_i) \neq 0\}$. Let $\tau = \mathsf{T}(g)$ for $g \in \mathcal{J}_{i-1}$.
- As before, $\tau \in \mathsf{N}(I(\{P_1, ..., P - i\}))$, so we add it to the Groebner escalier, removing it from the monomial basis.
- Set $\mathcal{J}_i = \mathcal{J}_{i-1} \setminus \{g\}$.
- Compute the triangular polynomial $q_i = \frac{1}{g(P_1)} g$.
- For each $f \in \mathcal{J}_{i-1}$ with $\mathsf{T}(f) > \tau$, interpolate in P_i: $f = f - f(P_i) q_i$. Substitute in $\mathcal{J}_i$ f with its new value.
- Compute the terms $x_j \tau$, for $x_j \leq \min(\tau)$ (5.7.1). We have at least one term, namely $\min(\tau)\tau$.
- Apply the subroutine GaussRed to the terms of the previous step.
- Insert in $\mathcal{J}_i$ the obtained polynomials.

If $i = S$, the algorithm stops and returns $\mathcal{J}_i = \mathcal{J}_S$. Otherwise, i is incremented by one and the steps above repeated.
We display now the pseudocode of this Moeller version.

We display now an example of the execution of algorithm 10.

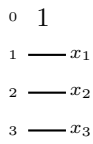
Algorithm 10 Involutive basis Moeller algorithm.

procedure JANM($\mathbf{X}$) $\to \mathcal{J}, \mathsf{N}, q$ ▷ $\mathcal{J}$ is the reduced involutive basis of I, N is the associated Groebner escalier and q is a triangular set for $\mathbf{X}$. Denote $\mathbf{X} = \{P_1, ..., P_S\}$, $P_i = (a_{i,1}, ..., a_{i,n})$, $i = 1, ..., n$.
  $\mathcal{J} = \{x_1 - a_{1,1}, ..., x_n - a_{1,n}\}$
  $\mathsf{N} = \{1\}$
  $q = \{1\}$ ▷ This is the output for the case $|\mathbf{X}| = 1$.
  for $i = 2$ to n **do**
    $\tau = \min\{\mathsf{T}(f),\ f \in \mathcal{J},\ f(P_i) \neq 0\}$
    $\mathsf{N} = \mathsf{N} \cup \{\tau\}$
    Let $f \in \mathcal{J}$ such that $\mathsf{T}(f) = \tau$
    $\mathcal{J} = \mathcal{J} \setminus \{f\}$
    $q = q \cup \{\frac{1}{f(P_i)} f\}$
    for each $f \in \mathcal{J}$ with $\mathsf{T}(f) > \tau$ **do**
      $f = f - f(P_i) q_i$
    end for
    for $j \leq \min(\tau)$ **do**
      $p = GaussRed(x_j \tau)$
      $\mathcal{J} = \mathcal{J} \cup \{p\}$
    end for
  end forreturn $\mathcal{J}, \mathsf{N}, q$.
end procedure

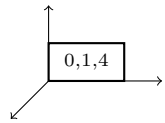
Example 6.8.1. We consider the set $\mathbf{X} = \{(0,1,4),(1,0,1),(0,2,0),(1,3,4),(0,3,2),(1,0,6)\} \in \mathbb{R}^3$ and we apply to it algorithm 10 in order to compute the reduced involutive basis of $I = I(\mathbf{X}) \triangleleft \mathbf{k}[x_1,x_2,x_3]$.
In order to clarify how the structure varies as we add a new point, we draw the Bar Code and the tower structure step by step.
As explained in the comments above, the first point $P_1 = (0,1,4)$ is associated to

- $\mathcal{J} = \{x_1, x_2 - 1, x_3 - 4\}$;
- $\mathsf{N} = \{1\}$;
- $q = \{1\}$.

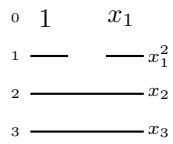
The Bar Code equipped with the star set is



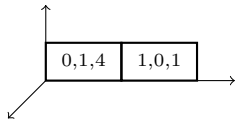
while the tower structure is



We take these data as base points for the procedure and we continue with $P_2 = (1,0,1)$.
Since the minimal polynomial in the current $\mathcal{J}$, not vanishing in P_2 is x_1, then $\mathsf{N} = \{1, x_1\}$ and $q_2 = x_1$. The Bar Code is



and the tower structure is

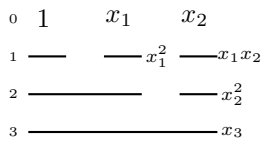


We remove x_1 from the involutive basis, so $\mathcal{J} = \{x_2 - 1, x_3 - 4\}$. We interpolate these polynomials using q_2 in order to make them vanish both in P_1 and in P_2:

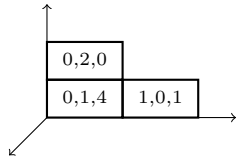
$x_2 - 1 \to x_2 - 1 - ev_{P_2}(x_2 - 1)q_2 = x_2 + x_1 - 1$;

$x_3 - 4 \to x_3 + 3x_1 - 4$;

so we get $\mathcal{J} = \{x_2 + x_1 - 1, x_3 + 3x_1 - 4\}$.
Since $\min(x_1) = x_1$, we only have to apply GaussRed only to the term x_1^2. Since x_1^2 already vanishes in P_1, we get $x_1^2 - x_1$, so $\mathcal{J} = \{x_1^2 - x_1, x_2 + x_1 - 1, x_3 + 3x_1 - 4\}$.
We consider now the point $P_3 = (0, 2, 0)$. Since $x_1^2 - x_1$ vanishes in P_3, while $ev_{P_3}(x_2 + x_1 - 1) = 1$, we get $\mathsf{N} = \{1, x_1, x_2\}$ and $q_3 = x_2 + x_1 - 1$.
The Bar Code is



the tower structure is



We get $\mathcal{J} = \{x_1^2 - x_1, x_3 + 3x_1 - 4\}$.
We interpolate the polynomial in x_3:

$x_3 + 3x_1 - 4 \to x_3 + 3x_1 - 4 - ev_{P_3}(x_3 + 3x_1 - 4)q_3 = x_3 + 4x_2 + 7x_1 - 8$.

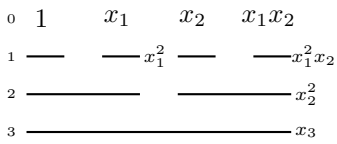
Since $\min(x_2) = x_2$, we add to the star set the terms x_1x_2, x_2^2, so we have to apply GaussRed to these terms:

$x_1x_2 \to x_1x_2$;

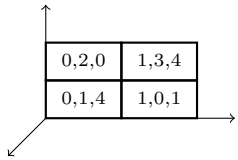
$x_2^2 \to x_2^2 - 3x_2 - 2x_1 + 2$.

Then we get $\mathcal{J} = \{x_1^2 - x_1, x_1x_2, x_2^2 - 3x_2 - 2x_1 + 2, x_3 + 4x_2 + 7x_1 - 8\}$.
Consider the point $P_4 = (1, 3, 4)$. The first polynomial non vanishing in P_4 is x_1x_2, so $\mathsf{N} = \{1, x_1, x_2, x_1x_2\}$ and $q_4 = \frac{1}{3}x_1x_2$.

The Bar Code is



the tower structure is

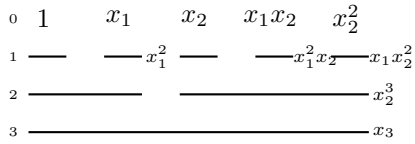


We have $\mathcal{J} = \{x_1^2 - x_1, x_2^2 - 3x_2 - 2x_1 + 2, x_3 + 4x_2 + 7x_1 - 8\}$, so we interpolate the last two polynomials:

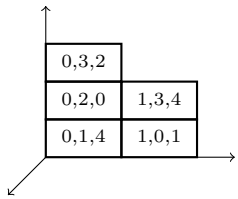
$$x_2^2 - 3x_2 - 2x_1 + 2 \to x_2^2 - 3x_2 - 2x_1 + 2$$

$$x_3 + 4x_2 + 7x_1 - 8 \to x_3 - 5x_1x_2 + 4x_2 + 7x_1 - 8,$$

then $\mathcal{J} = \{x_1^2 - x_1, x_2^2 - 3x_2 - 2x_1 + 2, x_3 - 5x_1x_2 + 4x_2 + 7x_1 - 8\}$.
Since $\min(x_1x_2) = x_1$, we have to deal with $x_1^2x_2$, obtaining $x_1^2x_2 - x_1x_2$, so $\mathcal{J} = \{x_1^2 - x_1, x_1^2x_2 - x_1x_2, x_2^2 - 3x_2 - 2x_1 + 2, x_3 - 5x_1x_2 + 4x_2 + 7x_1 - 8\}$.
We continue with $P_5 = (0, 3, 2)$. The first polynomial not vanishing in P_5 is $x_2^2 - 3x_2 - 2x_1 + 2$, so $\mathsf{N} = \{1, x_1, x_2, x_1x_2, x_2^2\}$ and $q_5 = \frac{1}{2}(x_2^2 - 3x_2 - 2x_1 + 2)$.
The Bar Code is



the tower structure is



We have $\mathcal{J} = \{x_1^2 - x_1, x_1^2x_2 - x_1x_2, x_3 - 5x_1x_2 + 4x_2 + 7x_1 - 8\}$ and we only have to interpolate the polynomial in x_3:

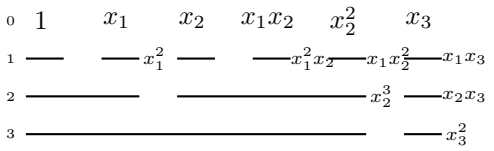
$$x_3 - 5x_1x_2 + 4x_2 + 7x_1 - 8 \to x_3 - 5x_1x_2 - 3x_2^2 + 13x_2 + 13x_1 - 14.$$

Since $\min(x_2^2) = x_2$, we apply GaussRed to $x_1x_2^2, x_2^3$:

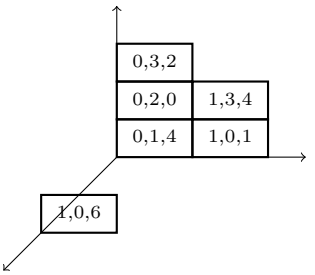
$$x_1x_2^2 \to x_1x_2^2 - 3x_1x_2;$$

$x_2^3 \to x_2^3 - 6x_2^2 - 2x_1x_2 + 11x_2 + 6x_1 - 6$,

so $\mathcal{J} = \{x_1^2 - x_1, x_1^2x_2 - x_1x_2, x_1x_2^2 - 3x_1x_2, x_2^3 - 6x_2^2 - 2x_1x_2 + 11x_2 + 6x_1 - 6, x_3 - 5x_1x_2 - 3x_2^2 + 13x_2 + 13x_1 - 14\}$.
At the end, we conclude with $P_6 = (1, 0, 6)$. All the polynomials in the current $\mathcal{J}$ vanish in P_6 but $x_3 - 5x_1x_2 - 3x_2^2 + 13x_2 + 13x_1 - 14$.
Thereby, the final Groebner escalier is $\mathsf{N} = \{1, x_1, x_2, x_1x_2, x_2^2, x_3\}$ and $q_6 = \frac{1}{5}(x_3 - 5x_1x_2 - 3x_2^2 + 13x_2 + 13x_1 - 14)$. Then $\mathcal{J} = \{x_1^2 - x_1, x_1^2x_2 - x_1x_2, x_1x_2^2 - 3x_1x_2, x_2^3 - 6x_2^2 - 2x_1x_2 + 11x_2 + 6x_1 - 6\}$.
The Bar Code is



the tower structure is



In order to get the involutive basis we only have to perform GaussRed to x_1x_3, x_2x_3, x_3^2:

$x_1x_3 \to x_1x_3 - x_3 + 4x_1x_2 + 3x_2^2 - 13x_2 - 14x_1 + 14$;

$x_2x_3 \to x_2x_3 - 5x_2^2 - 8x_1x_2 + 19x_2 + 18x_1 - 18$;

$x_3^2 \to x_3^2 - 7x_3 + 11x_2^2 + 14x_1x_2 - 45x_2 - 4x_1 + 46$.

The output is then

- $\mathsf{N} = \{1, x_1, x_2, x_1x_2, x_2^2, x_3\}$;
- $q = \{1, x_1, x_2+x_1-1, \frac{1}{3}x_1x_2, \frac{1}{2}(x_2^2-3x_2-2x_1+2), \frac{1}{5}(x_3-5x_1x_2-3x_2^2+13x_2+13x_1-14)\}$;
- $\mathcal{J} = \{x_1^2 - x_1, x_1^2x_2 - x_1x_2, x_1x_2^2 - 3x_1x_2, x_2^3 - 6x_2^2 - 2x_1x_2 + 11x_2 + 6x_1 - 6, x_1x_3 - x_3 + 4x_1x_2 + 3x_2^2 - 13x_2 - 14x_1 + 14, x_2x_3 - 5x_2^2 - 8x_1x_2 + 19x_2 + 18x_1 - 18, x_3^2 - 7x_3 + 11x_2^2 + 14x_1x_2 - 45x_2 - 4x_1 + 46\}$.

The correctness of the algorithm is a straightforward consequence of the one of the original Moeller algorithm and from proposition 5.7.1 on the variations of the star set when we add a term to the Groebner escalier.

Part IV

The Axis of Evil Theorem applied to error correcting codes.

CHAPTER 7

Error correcting codes and locator polynomials.

7.1 Introduction.

Coding theory is a rather recent subject. As a matter of fact, it dates back to 1948, with an illuminating paper by Claude Elwood Shannon [96], which originated both coding theory and information theory.
In this chapter we recall the notions on error correcting codes, needed to understand the joint work with Massimiliano Sala and Teo Mora, examined in chapter 8, which links the Axis of Evil Theorem to error correcting codes.
First of all, we introduce the notion of code and some preliminary definitions.
Starting with the so called *Cooper's philosophy* [28, 29], going on with Chen's works [24, 25] and with the papers by Teo Mora, Emanuela Orsini and Massimiliano Sala [77, 82, 83], the idea of exploiting Groebner bases computations in order to decode cyclic codes gained around and became more and more important.

In these works, given some well determined set of polynomials, the lexicographical reduced Groebner basis is computed and employed for the decoding process, in order to detect and correct the errors eventually occurred during a transmission, by making some computations with the so called *locator polynomials*.
Being rather complicated to get the errors from the syndromes, Cooper has the idea to turn the problem into a problem on polynomials. More precisely, Cooper takes a (finite) set of polynomials $\mathcal{F}_C$, such that the error locations are in $V(\mathcal{F}_C)$ and he computes the lexicographical reduced Groebner basis of $I = (\mathcal{F}_C)$. The required error locator polynomial can be directly computed via the elimination property of lexicographical Groebner bases.
Chen et al. developed Cooper's theory, following two directions. More precisely:

- they gave an approach to decoding via Newton identities, which was improved by Augot-Bardet-Faugere [3, 4];
- they introduced the so called *syndrome variety* and the related *syndrome* ideal and proposed to deduce via a Groebner basis pre-computation a series of polynomials from which they deduce the plain error locator polynomial for each error and associated syndromes. This approach has been refined by Loustaunau and York [66] and Caboara-Mora [13].

The investigation on the structure of the syndrome variety and on its Groebner basis shows that most of its roots are spurious [23] and that the pre-computed polynomials have *telescopical relations* [6, 13].
Finally, Orsini and Sala [78] improved the decoding process by eliminating the *spurious solutions* of the system and introduced the *general error locator polynomial*.
In further investigations (in cooperation with Teo Mora) [82, 83], they also highlighted the importance for the general error locator polynomial to be sparse: this is the main link with our work (chapter 8).
In the first section, we recall the basic concepts of coding theory, starting with the communication channel model proposed by Shannon. In sections 7.3 and 7.4 we deal with *linear codes* and a peculiar typology of linear codes, called *cyclic codes*, showing their main features.
In section 7.5, we introduce Cooper's philosophy and the developments proposed in the following years.

7.2 A glimmer of error correcting codes.

It is possible to declare that both *coding theory* and *information theory* date back to the milestone paper by Claude Elwood Shannon "A mathematical theory of communication", pub-

lished in 1948 [96].
In this paper, the author describes a scheme of a *communication channel*, as in the following picture 7.1.

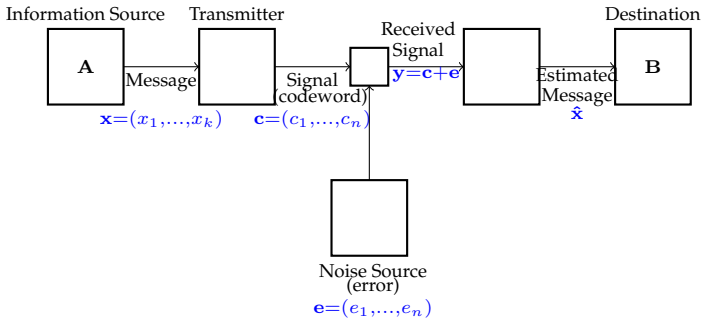


Figure 7.1: The communication channel by C.E. Shannon.

A communication channel consists five different parts.

- The *Information source*: is the source producing the message to be sent to a receiving terminal. As remarked by Shannon, the message can be of various types, such as a sequence of letters as in a telegraph or a single function of time $f(t)$ as occurs for radio or telephony.
- The *Transmitter*: is the device operating on the message, *encoding* it, in order to produce a *signal*, which is suitable for the transmission on the channel.
- The *Channel*: is the medium used to transmit the signal from the transmitter to the receiver. For example a channel can be a band of radio frequencies or a cable.
- The *Receiver*: is the device performing the inverse operation of the transmitter. More precisely, it *decodes* the signal, extracting the message from it.
- The *Destination*: is the person or thing to which the message is intended.

The channel can be *noisy*, i.e. when the information passes through it, there can be some interference.
The encoding procedure is an injective map from the space containing the possible messages to a larger space. Roughly speaking, it adds some redundancy to the given message, lengthening it.

On the other hand, the decoding procedure recovers the original message.

For simplicity's sake, from now on we assume (as it is usually done) the encoding to be a linear function between two vector spaces.
In the next sections, we give an overall view of the fundamentals of coding theory, loosely following [53, 77].

7.3 Linear codes.

Linear codes have been deeply studied, since they have algebraic properties making them much easier to describe than the non-linear ones.
We denote by $\mathbb{F}_q := GF(q)$, with $q = p^m$ and p a prime number, the finite field of cardinality q and we write $(\mathbb{F}_q)^n$ for the vector space constituted by the n-tuples of the elements in $\mathbb{F}_q$, which are regarded as row vectors.

Definition 7.3.1. Given $k, n \in \mathbb{N}$, such that $1 \leq k \leq n$, a *linear code* C is a vector subspace of $(\mathbb{F}_q)^n$ of dimension k.
We say that C is a linear code over $\mathbb{F}_q$ of length n and dimension k, for short $[n, k]_q$ code.
A vector $\mathbf{c} \in C$ is named *codeword* or *word* for short.

The codewords are indifferently denoted by

$$\mathbf{c} = (c_1, ..., c_n) = c_1 c_2 ... c_n.$$

Each c_i, $i = 1, ..., n$ is called *symbol*.
We define the usual scalar product on $(\mathbb{F}_q)^n$ and we denote it by "$\cdot$". This way, if $C \subset (\mathbb{F}_q)^n$ is a vector subspace, then we can define the dual vector space $C^\perp$ and then we can talk about *dual codes*.

Definition 7.3.2. If C is an $[n, k]_q$ code, its *dual code* is the set $C^\perp$, containing the vectors orthogonal to all the words in C, i.e.

$$C^\perp = \{c' \in (\mathbb{F}_q)^n,\ c' \cdot c = 0,\ \forall c \in C\}.$$

The dual code of an $[n, k]_q$ code is clearly an $[n, n-k]_q$ code.

Definition 7.3.3. A *generator matrix* of an $[n, k]_q$ code C is a $(k \times n)$-matrix G whose rows form a basis of C as a $\mathbb{F}_q$-vector space.

An $[n,k]_q$ code C, in general, has more than one generating matrix.
If $G = (I_k|A)$, where I_k is the $(k \times k)$-identity matrix, G is a *generator matrix in standard form*.
Given a generator matrix G of the given $[n,k]_q$ code C, any set of k independent columns of G corresponds to a set of coordinates, forming the so called *information set* of C.
The remaining $r = n - k$ coordinates form the *redundancy set* of C, while r is its *redundancy*.
The encoding of a linear code is very simple. Given a message $\mathbf{m} \in (\mathbb{F}_q)^k$ and a generator matrix G, we can obtain the word $\mathbf{c} \in (\mathbb{F}_q)^n$ by simple matrix multiplication $\mathbf{c} = \mathbf{m}G$.
When G is a generator matrix in standard form we get $\mathbf{c} = (\mathbf{m}, \mathbf{m}A)$: the message $\mathbf{m}$ is composed by the first k components of $\mathbf{c}$. Such an encoding is called *systematic*.

Definition 7.3.4. A *parity-check matrix* for an $[n,k]_q$ code C is a generator $((n-k)\times n)$-matrix H for $C^\perp$.

We can represent a linear code C exploiting the parity-check matrix H:

$$\forall \mathbf{x} \in (\mathbb{F}_q)^n,\ H^t\mathbf{x} = 0 \Leftrightarrow \mathbf{x} \in C.$$

Let us now briefly describe a transmission process. Suppose one has to send the message $\mathbf{x} \in (\mathbb{F}_q)^k$. The transmitted word is then $\mathbf{c} = \mathbf{x}G \in (\mathbb{F}_q^n)$.
Let $\mathbf{y} \in (\mathbb{F}_q)^n$ the received n-tuple. Due to the interference peculiar to the channel, there are exactly four possibilities which can come up:

a. $\mathbf{y} = \mathbf{c} \in C$: the receiver deduces (correctly) that no errors have occurred during the transmission and recovers the message as $\mathbf{x}$.

b. $\mathbf{y} \notin C$: the receiver is able to deduce that some error has occurred. It detects and corrects the errors by supposing that the correct word is the one in C differing from $\mathbf{y}$ in the *minimal number of positions*.

c. $\mathbf{y} \notin C$: again the receiver is able to deduce that some error has occurred, but if it tries a correction as in b. it gets another codeword, different from $\mathbf{c}$ and so it gets a wrong message.

d. $\mathbf{y} \in C$, but $\mathbf{y} \neq \mathbf{c}$: in this case, the receiver believes no errors have occurred and it is completely wrong.

In order to correct the errors, the receiver needs to find the codeword having the "highest possibility" of been sent by the transmitter, so it needs to understand *how the noise can affect the transmitted word*.

Definition 7.3.5 ([77])**.** A q-ary *symmetric channel*, denoted by SC from now on, is a channel satisfying the conditions below:

1. the component of a transmitted word (an element of $\mathbb{F}_q$ that here we name generally "symbol") can be changed by the noise only to another element of $\mathbb{F}_q$;

2. the probability that a symbol becomes another one is the same for all pairs of symbols;

3. the probability that a symbol changes during the transmission[1] does not depend on its position;

4. if the i-th component is changed, then this fact does not affect the probability of change for the j-th components, even if j is close to i.

In his paper [96], Shannon considers a channel with input alphabet $a_1, ..., a_k$ and output alphabet $b_1, ..., b_l$, supposing that each output letter depends statistically on the corresponding input letter only according to a fixed probability. We write $\mathbb{P}(b_j|a_i)$ for the probability that b_j is received if a_i is transmitted.
Such a channel is called *discrete memoryless channel*, DMC for short.
In particular, he deals with binary symmetric codes [105].
If we take a binary code of k words of length n (we choose k out of 2^n words), we say that the *information rate* is $R = n^{-1}log_2(k)$.
Consider a binary symmetric code with error probability $0 < p < \frac{1}{2}$ and suppose to have a code consisting of M vectors, chosen in $\{0,1\}^n$, with some decoding rule. Denote by $\mathbb{P}_i$ the probability that an error occurs, after decoding, if $x_i \in M$ is transmitted. The probability of error when using this code is

$$\mathbb{P}_{error} = M^{-1} \sum_{i=1}^{M} \mathbb{P}_i.$$

We define $\mathbb{P}^*(M, n, p)$ as the minimum of $\mathbb{P}_{error}$ over all codes with the given parameters.
The capacity of the binary symmetric code is $C = 1 + p \cdot log(p) + (1-p)log(1-p)$.
We state now *Shannon's fundamental theorem*.

Theorem 7.3.6. Let $M_n := 2^{[Rn]}$, where $0 < R < C$. Then $P^*(M_n, n, p) \to 0$ if $n \to \infty$.

This means that there is a sequence of codes with information rate tending to R and error probability tending to 0. In other words, given $\epsilon > 0$ and $R < C$ there is a code with rate $> R$ and error probability $< \epsilon$.
From now on, we assume to have a SC, such that *all the words are sent with the same probability* and that *the probability of a symbol to be corrupted is less than the one of being maintained unchanged by the interference.*

[1] It is the error probability, namely the probability of an error to occur.

Actually, this assumption is merely theoretical, since it is not reasonable in practice. Anyway, it allows a simple construction of the theory, so it is classically accepted.
Under our hypotheses, we can construct a "good code" separating the codewords inside $(\mathbb{F}_q)^n$ as much as possible and this leads to the following

Definition 7.3.7. Given two elements $\mathbf{v}, \mathbf{w} \in (\mathbb{F}_q)^n$, the *Hamming distance* of $\mathbf{v}, \mathbf{w}$ is the number $d_H(\mathbf{v}, \mathbf{w})$ of coordinates in which they differ.

Definition 7.3.8. The *Hamming weight* of $\mathbf{v} \in (\mathbb{F}_q)^n$ is the number of its nonzero coordinates, i.e. $w(\mathbf{u}) := d_H(\mathbf{u}, \mathbf{0})$.

Definition 7.3.9. The *distance* of a code C is the minimal distance between two distinct words

$$d_H(C) := \min\{d_H(\mathbf{v}, \mathbf{w}) | \, \mathbf{v}, \mathbf{w} \in C, \, \mathbf{v} \neq \mathbf{w}\}.$$

Given an $[n, k]_q$ code C, we denote it by $[n, k, d]_q$ code if $d = d_H(C)$.
The distance is very important for a linear code, since it allows to compute two fundamental numbers:

- the *error detection capability*, i.e. the number of errors that the code can detect;
- the *error correction capability*, namely the number of errors that the code can correct.

Theorem 7.3.10 ([77])**.** If C is an $[n, k, d]_q$ code it has

- error detection capability $l = d - 1$;
- error correction capability $t = \lfloor \frac{(d-1)}{2} \rfloor$.

From now on, we denote t the error correction capability of a code C.

Theorem 7.3.11 (Singleton Bound, [77])**.** Given an an $[n, k, d]_q$ code C it holds

$$d \leq n - k + 1.$$

Each code for which equality holds is called *minimum distance separable code* or simply MDS.

Proposition 7.3.12 ([52])**.** If the employed code is SC with error correction capability t and the probability of a symbol to be corrupted is less than the one of being maintained unchanged by the interference, then the sent codeword with the highest probability is the one nearest w.r.t. Hamming distance to the received vector. Such a codeword is unique if no more than t errors have occurred.

Consider an $[n, k]_q$ code C and let

- $\mathbf{c} \in (\mathbb{F}_q^n)$ the transmitted word;
- $\mathbf{e} \in (\mathbb{F}_q^n)$ the occurred error;
- $\mathbf{y} \in (\mathbb{F}_q^n)$ the received vector.

It holds

$$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

and, given $\mathbf{y}$, we want to find an $\mathbf{e}$ of *minimal weight* such that $\mathbf{y} - \mathbf{e} \in C$. For this purpose, we consider the parity check matrix H and we have

$$H^t\mathbf{y} = H^t(\mathbf{c} + \mathbf{e}) = H^t(\mathbf{e}) = \mathbf{s} \in (\mathbb{F}_q^{n-k}).$$

Definition 7.3.13. All the elements of the form $\mathbf{s} = H^t\mathbf{y} \in (\mathbb{F}_q^{n-k})$ are called *syndromes*. In particular, we say that $\mathbf{s}$ is the *syndrome corresponding to* $\mathbf{y}$.

We point out that the syndrome depends only on the occurred error, not on the transmitted word.
If $\mathbf{v} \in (\mathbb{F}_q)^n$ we define its associated *coset* as

$$\mathbf{v} + C = \{\mathbf{v} + \mathbf{c} | \, \mathbf{c} \in C\}.$$

We get:

$$\mathbf{v}, \mathbf{w} \in (\mathbb{F}_q)^n \text{ are in the same coset } \Leftrightarrow \mathbf{v} - \mathbf{w} \in C.$$

The given vector space $(\mathbb{F}_q)^n$ can be partitioned into q^{n-k} cosets of size q^k.

Proposition 7.3.14. Given an $[n, k, d]_q$ code C, $\mathbf{v}, \mathbf{w} \in (\mathbb{F}_q)^n$ belong to the same coset if and only if they have the *same syndrome*.

Definition 7.3.15. Given a coset $\mathbf{v} + C$ of an $[n, k, d]_q$ code C and a vector $\mathbf{w} \in \mathbf{v} + C$, $\mathbf{w}$ is a *coset leader* if it is an element of *minimum weight* in $\mathbf{v} + C$.

With the previous notation, we define *correctable syndromes*.

Definition 7.3.16. If s is a syndrome corresponding to an error of weight $\mathbf{w}(s) \leq t$, then we say that s is a *correctable syndrome*.

Theorem 7.3.17 (Correctable syndromes, [77])**.** If C is an $[n, k]_q$ code with error correction capability t and the occurred errors are in number *smaller or equal* then t, then there exists *only one error* $\mathbf{e}$ corresponding to the correctable syndrome $\mathbf{s} = H\mathbf{e}$ and $\mathbf{e}$ is the *unique coset leader* of $\mathbf{e} + C$.

Now we are ready to *decode a linear code*.
Given the received vector $\mathbf{y} \in (\mathbb{F}_q)^n$, we first compute the syndrome $\mathbf{s} = H\mathbf{y}$. Then we find a coset leader for the coset associated to $\mathbf{s}$ (7.3.14), say $\mathbf{z}$. The decoded word is $\mathbf{c} = \mathbf{y} - \mathbf{z}$ and we only have to recover the message from it.
In order to perform the procedure above, we need to construct a matrix, called *standard array*, containing all the vectors in $(\mathbb{F}_q)^n$, which are 2^n, ordered by coset. We can conclude that the complexity of the decoding procedure is exponential in terms of memory occupancy.
Both the problem of decoding a linear code and the general problem of finding the distance of a linear code are NP-complete, as shown in [5, 7, 106]. There are no algorithms decoding linear codes in polynomial time.

7.4 Cyclic codes.

In this thesis we will deal with some peculiar codes, called *cyclic codes*.

Definition 7.4.1. An $[n, k, d]_q$ code C is called *cyclic* if

$$(c_0, ..., c_{n-1}) \in C \Leftrightarrow (c_1, ..., c_{n-1}, c_0) \in C.$$

Essentially, definition 7.4.1 says that a cyclic permutation of the components of a word gives again a word of C.
Cyclic codes can be algebraically described through a *polynomial representation for words*. More precisely, denoted by $\mathbb{F}_q[x]$ the polynomial ring in one variable with coefficients in the finite field $\mathbb{F}_q$, we consider the principal ideal $I = (x^n - 1) \lhd \mathbb{F}_q[x]$ and the quotient $\mathcal{R} := \mathbb{F}_q[x]/I$ and we construct the following bijection

$$Wp : (\mathbb{F}_q)^n \to \mathcal{R}$$

$$\mathbf{v} = (v_0, ..., v_{n-1}) \mapsto v_0 + ...v_{n-1}x^{n-1}.$$

Thanks to Wp, we can view a linear code as a subset of $\mathcal{R}$; in the following theorem, we characterize cyclic codes.

Theorem 7.4.2. An $[n, k, d]_q$ code C is cyclic if and only if C is an ideal of the quotient ring $\mathcal{R}$.

Being $\mathcal{R}$ a PIR, for each C, there is a *unique* monic polynomial generating it, the *generator polynomial* g of C.

It holds $deg(g) = n - k$ and $g \mid x^n - 1$. Using $g = \sum_{i=0}^{n-k} g_i x^i$ one can recover a generator matrix for the code:

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}$$

Moreover, given $f \in \mathcal{R}$, we have

$$f \in C \Leftrightarrow \exists q \in \mathcal{R} | \, f = qg.$$

In analogy with matrices, we can define the *parity check polynomial* of C from the generator polynomial.
Actually, $g \mid (x^n - 1)$ and it is unique, so the parity check polynomial is simply the polynomial $h \in \mathcal{R}$ such that

$$h(x) = \frac{x^n - 1}{g(x)}$$

and, for $f(x) \in C$, we have

$$f(x) \in C \Leftrightarrow f(x) = q(x)g(x) \Leftrightarrow f(x)h(x) = q(x)(g(x)h(x)) = 0 \text{ in } \mathcal{R}.$$

We remark also that the generator polynomial of the dual code $C^\perp$ is $g^\perp(x) = x^{deg(h)} h(x^{-1})$ (c.f. [77]).
We deal now with the problem of encoding and decoding, given an $[n, k, d]_q$ code C with generator polynomial g, which allows to encode q-ary messages of length k by adding $n - k$ symbols as redundancy.
Let then $\mathbf{m} = (m_0, ..., m_{k-1})$ a message and consider the associated $m(x) = \sum_{i=0}^{k-1} m_i x^i \in R$. We can obtain a systematic encoding for $m(x)$. For this purpose, we multyply $m(x)$ by x^{n-k} and we divide the result by $g(x)$, getting

$$m(x)x^{n-k} = q(x)g(x) + r(x)$$

with $deg(r(x)) < deg(g(x)) = n - k$, so the reminder $r(x)$ can be viewed as an $(n-k)$-vector. Joining the k-vector m with the (nk)-vector r we obtain an n- vector c, which is the encoded word, i.e.

$$c(x) := m(x)x^{n-k} - r(x).$$

Therefore, the decoding process is immediate, if no errors occur, since the message is constituted by the last k components of the received vector.

When the receiver gets a vector and has to check the presence or absence of errors, only has to check whether the remainder of the division of the polynomial associated to the received vector by g is equal to zero to state that "probably" no errors have occurred. If the reminder is not zero, it gives the syndrome, so the error can be corrected in the same way as described in the previous section.
Given $\mathbb{F}_q$, we have $x^n - 1 = \prod_{j=1}^{r} f_j$, f_j irreducible over the base field.
Since cyclic codes of length n over $\mathbb{F}_q$ are generated by divisors of $x^n - 1$, each of these codes corresponds to a subset of $\{f_j\}_{j=1}^{r}$.
In particular, let us assume $GCD(n, q) = 1$, $\mathbb{F}_{q^m}$ the splitting field of $x^n - 1$ over $\mathbb{F}_q$ and a a primitive n-th root of unity over $\mathbb{F}_q$. Clearly

$$x^n - 1 = \prod_{i=0}^{n-1} (x - a^i)$$

and the generator polynomial of G has, as roots, some powers of a

Definition 7.4.3. The *complete defining set* of an $[n, k, d]_q$ cyclic code C with $GCD(n, q) = 1$ and generator polynomial g_C is

$$S_{C,a} := S_C = \{i_1, ..., i_{n-k} | g_C(a^{i_j}) = 0,\ j = 1, ..., n - k\}$$

From now on, we fix a primitive n-th root of the unity a and we always write S_C instead of $S_{C,a}$.
We can collect the integers modulo n into q-cyclotomic classes C_i:

$$\{1, ..., n - 1\} = \bigcup C_i,\ C_i = \{1, qi, ..., q^r i\},$$

where r is the smallest integer such that $i \cong iq^r \bmod n$.
The complete defining set is then a collection of q-cyclotomic classes. For this reason, there are some $S_{C'} \subset S_C$ which are sufficient to specify the code unambiguously. We call each of them *defining set*.
Some special cyclic codes are the so called *BCH codes*, which allow decoding procedures that are faster than the one sketched above (see [77] for more details).

Theorem 7.4.4 (BCH bound)**.** Consider an $[n, k, d]_q$ cyclic code C, with $GCD(n, q) = 1$ and defining set $S_C = \{i_1, ..., i_{n-k}\}$. Suppose there are $\delta - 1$ consecutive number in S_C, say $\{m_0 + i,\ 0 \leq i \leq \delta - 2\} \subset S_C$. Then $d \geq \delta$.

Definition 7.4.5. If C is the $[n, k, d]_q$ cyclic code, with defining set $S = \{m_0 + i,\ 0 \leq i \leq \delta - 2,\ m_0 \geq 0,\ m_0 + \delta - 2 \leq n - 1\}$, then C is a *BCH code* of *designed distance* δ.
A BCH code is *narrow sense* if $m_0 = 1$ and *primitive* if $n = q^m - 1$.

There are several methods in order to decode a BCH code. For example, we can use the *extended Euclid algorithm*.
We consider a BCH code of length n over $\mathbb{F}_q$, with error correction capability t and designed distance $\delta = 2t + 1$ and we denote by a a primitive n-th root of unity in $\mathbb{F}_{q^m}$.
We take a word $c(x) = c_0 + ... + c_{n-1}x^{n-1}$ and we denote by $v(x) = v_0 + ... + v_{n-1}x^{n-1}$ the received word.
We can represent the error vector as the *error polynomial*

$$e(x) = e_0 + ... + e_{n-1}x^{n-1}.$$

If $\mu \leq t$ is the weight of the error, let $L = \{l | e_l \neq 0, 0 \leq l \leq n-1\}$ be the *set of the error positions* and $\{a^l | l \in L\}$ the set of *error locators*. We call *error values* the values e_l, $l \in L$. The *classical error locator polynomial* is

$$\sigma(x) = \prod_{l \in L}(1 - xa^l),$$

but we can recover the error locations also using the *plain error locator polynomial*, i.e.

$$L_e(x) = \prod_{l \in L}(x - a^l).$$

The *error evaluator polynomial* is

$$\omega(x) = \sum_{l \in L} e_l a^l \prod_{i \in L \setminus \{l\}} (1 - xa^i).$$

In order to correct the errors, we find $\sigma(x)$ and $\omega(x)$:

an error is in position l if and only if $\sigma(a^{-l}) = 0$ and in this case the value of the error is

$$e_l = -a^{-l}\frac{\omega(a^{-l})}{\sigma'(a^{-l})},$$

where $\sigma'(x)$ is the first derivative of $\sigma(x)$.

Lemma 7.4.6. The polynomials $\sigma(x)$, $\omega(x)$ defined above are coprime.

In order to decode the given BCH code, we first compute the syndrome of the received vector $v(x)$:

$$H^t v = \begin{pmatrix} 1 & a & a^2 & \cdots & a^{n-1} \\ 1 & a^2 & a^4 & \cdots & a^{2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & a^{\delta-1} & a^{2(\delta-1)} & \cdots & a^{(\delta-1)(n-1)} \end{pmatrix} \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{pmatrix} = \begin{pmatrix} e(a) \\ e(a^2) \\ \vdots \\ e(a^{\delta-1}) \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_{2t} \end{pmatrix}.$$

The *syndrome polynomial* is $S(x) = \sum_{i=1}^{2t} S_i x^{i-1}$, with $S_i = \sum_{l \in L} e_l a^{il}$.

Theorem 7.4.7 (Key equation, [77]). For the polynomials $\sigma(x), \omega(x)$ the *key equation* holds

$$\sigma(x)S(x) \cong \omega(x) \bmod x^{2t}.$$

If there are polynomials $\sigma'(x), \omega'(x)$ with $deg(\omega'(x)) < deg(\sigma'(x)) \leq t$, satisfying the key equation, then there is a polynomial $\lambda(x)$ such that $\sigma'(x) = \lambda(x)\sigma(x)$ and $\omega'(x) = \lambda(x)\omega(x)$

The decoding algorithm consists essentially of finding $\sigma(x)$ and $\omega(x)$, availing of the key equation and the extended Euclid algorithm and Bézout identity [77].
Once noticed that $deg(\sigma(x)) \leq t$ and $deg(\omega(x)) \leq t-1$, we divide the polynomial $f(x) := x^{2t}$ and $g(x) = S(x)$ using the extended Euclid algorithm, denoting the reminder at each step h by $d_h(x)$. We stop when we find a $d_{k-1}(x)$ and $d_k(x)$ such that $deg(d_{k-1}(x)) \geq t$ and $deg(d_k(x)) \leq t-1$. Then, applying the procedure for Bézout identity, we get

$$d_k(x) = x^{2t}u_k(x) + S(x)v_k(x),$$

with $deg(v_k(x)) = deg(x^{2t}) - deg(d_{k-1}(x)) \leq 2t - t = t$.

Theorem 7.4.8. With the above notation, it holds $\sigma(x) = \lambda v_k(x)$ and $\omega(x) = \lambda d_k(x)$ for some $\lambda \in \mathbb{F}_q$.

We have $\lambda = v_k(0)^{-1}$, so that $\sigma(x) = \frac{v_k(x)}{v_k(0)}$ and $\omega(x) = \frac{d_k(x)}{v_k(0)}$.
Finally, if one wants to compute the error values, he can simply use the relations

$$e_l = -a^l \frac{\omega(a^{-l})}{\sigma'(a^{-l})},\ i = 1, ..., \mu.$$

We point out that we can also decode a BCH code using Berlekamp-Massey algorithm [6] or the so called *Cooper's philosophy*, explained in next section.

7.5 Cooper's philosophy and further improvements.

In his papers [28, 29], Cooper suggested to employ Groebner basis theory in order to decode cyclic codes.
More precisely, he considers a primitive binary BCH code of length $n = 2^m - 1$.
Let $a \in \mathbb{F}_{2^m}$ a primitive n-th root of unity and C our primitive BCH code over $\mathbb{F}_2$, with defining set $S_{C'} = \{2i+1,\ i = 0, ..., t-1\}$.

The related complete defining set is the union $S_C = \bigcup_{i=0}^{t-1} C_{2i+1}$, so it contains all the odd numbers from 1 to $2t-1$. Each even number $1 < \alpha < 2t-1$ is in the set, since $\alpha = 2^l h$

for some odd number $h < 2t-1$ and so $\alpha \in C_h$. This means that all the numbers from 1 to $2t-1$ are in S_C. Moreover $2t \in C_t \subset S_C$ and so we have at least $2t$ consecutive elements in S_C and the designed distance is $\delta \geq 2t+1$.
By the BCH bound (7.4.4), the distance is $d \geq 2t+1$ and the error correction capability turns out to be $t \geq \lfloor \frac{(\delta-1)}{2} \rfloor$.
Once received $\mathbf{v} \in (\mathbb{F}_2)^n$, the decoder computes the syndrome (7.3.13) $\mathbf{s} = (s_0, ..., s_{2t-1}) \in (\mathbb{F}_{2^m})^{2t}$, in order to find the error location a^j.
We define new variables $z_1, ..., z_t$, standing for the t error locations a^{l_i}, $l_i \in L$. Then, the error locations are a solution $(\xi_1, ..., \xi_t) \in (\mathbb{F}_{2^m})^t$ of a system of t polynomials over $\mathbb{F}_{2^m}[z_1, ..., z_t]$, i.e.

$$\mathcal{F}_C = \{f_i : \sum_{j=1}^{t} z_j^{2i-1} - s_{2i-1},\ i = 1, ..., t\}.$$

The problem for this nonlinear system is that sometimes is ineffective to compute its solutions, so Cooper proposes to find another simpler system, with the same solutions. Let then $I = (\mathcal{F}_C) \lhd \mathbb{F}_{2^m}[z_1, ..., z_t]$, $V(I)$ the defined variety, $\mathcal{G}$ the reduced Groebner basis of I, w.r.t. the lexicographical ordering, induced by $z_1 < ... < z_t$ and $g \in \mathbb{F}_{2^m}[z_1]$ the *unique* polynomial such that $\mathcal{G} \cap \mathbb{F}_{2^m}[z_1] = \{g\}$. We state here Cooper's theorem

Theorem 7.5.1 ([29])**.** If $\mathsf{E} = \{\xi_1, ..., \xi_\mu\}$ is the set of *error locations* and

$$\mathsf{Z} = \{\xi | (\xi, b_2, ..., b_t) \in V(I)\}$$

contains the components of all solutions of $\mathcal{F}_C$, then

- $\mathsf{E} = \mathsf{Z} = \{\xi | g(\xi) = 0\}$;
- $|\mathsf{E}| = \mu = deg(g) \leq t$;
- g is the polynomial whose roots are the error locators;
- $\sigma(z) = z^\mu g(z^{-1})$

In [24], Chen et al. generalize Cooper's idea to use Groebner techniques to binary cyclic codes.

They consider a binary cyclic code C with length n and defining set S. We denote by μ the number of occurred errors and v an integer such that $0 < v \leq t$ and $\mu \leq v$. Then, using the z_j's variables for the error locations[2], we can consider the following system where each

[2]They are n-th root of unity

syndrome $s_i \in \mathbb{F}_{2^m}$ represents a value:

$$\mathcal{F}_{CRHT_2} := \{\sum_{j=1}^{v} z_j^i - s_i,\ i \in S\} \cup \{z_j^{n+1} - z_j,\ 1 \leq j \leq v\} \subset \mathbb{F}_{2^m}[z_1, ..., z_v].$$

Such a system defines an ideal $I = (\mathcal{F}_{CRHT_2}) \triangleleft \mathbb{F}_{2^m}[z_1, ..., z_v]$, whose zero set gives the error locations and the error vector, occurred in the transmission. We look for it using Groebner bases .

Proposition 7.5.2. With the above notation

- $\mathsf{E} \subset \mathsf{Z} = \{\xi | g(\xi) = 0\}$;
- $|\mathsf{E}| = \mu \leq v = deg(g)$.

Theorem 7.5.3 ([24])**.** It holds:

1. If $v = \mu$, $V(I)$ consists of all coordinate permutations of $(\xi_1, ..., \xi_\mu)$, $\mathsf{E} = \mathsf{Z}$, $L_e(z) = g(z)$, $\sigma(z) = z^\mu g(z^{-1})$
2. If $v = \mu + 1$, $(0, \xi_1, ..., \xi_\mu) \in V(I)$, $\mathsf{E} = \mathsf{Z} \cup \{0\}$ and $g(z) = z(z^\mu \sigma(z^{-1})) = zL_e(z)$.
3. If $v \geq \mu + 2$ then $(\zeta, \zeta, \xi_1, ..., \xi_\mu, 0, ..., 0) \in V(I), \forall \zeta \in \mathbb{F}_{2^m}$, $\mathsf{E} = \mathbb{F}_{2^m}$ $g(z) = z^{n+1} - z$
4. If $v < \mu$ $\mathcal{G} = \{1\}$.

In [23] Chen et al. generalize Cooper's philosophy to q-adic codes proposing a solution for decoding an error whose weight is assumed known.
Moreover, they give an alternative approach via Newton's identities in the binary case, but, since it goes beyond our interest, we do not treat it. For details, one can see [77]. For the improvements by Augot-Bardet-Faugere, one can see [3, 4].
In the context defined so far, for any word to be decoded, we need to compute a Groebner basis and the syndromes are considered as parameters, computed expressively from the received word and substituted into the system. Moreover, different Groebner basis computations must be performed for different potential error weights, until the true weight of the actual error is obtained.
In [25], Chen et al. proposed a new method which consists of *considering the syndromes as variables* x_i and computing the Groebner basis as a preprocessing. The growth of the number of variables is a problem of this method. On the other hand, the Groebner basis is computed *only once*.
Following [77], we denote by $\mathbf{x}, \mathbf{y}, \mathbf{z}$ the multivariables representing, respectively, the syndromes, the locations and the error values, i.e. the variables for the polynomial ring

$\mathbb{F}_q[x_1, ..., x_{n-k}, z_t, ..., z_1, y_1, ..., y_t] = \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}]$.
Then, we consider

$$\mathcal{F}_{CRHT} := \{\sum_{j=1}^{t} y_j z_j^i - x_i,\ i \in S\} \cup \{z_j^{n+1} - z_j,\ 1 \leq j \leq t\} \cup \{y_j^{2^m-1} - 1,\ 1 \leq j \leq t\} \subset \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}],$$

$I = I(\mathcal{F}_{CRHT}) \lhd \mathbb{F}_{2^m}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$, $V(I) \subset (\mathbb{F}_{q^m})^{2\mu}$ and $\mathcal{G}$ the lexicographical reduced Groebner basis with $x_1 < ... < x_{n-k} < z_t, ..., z_1 < y_1 < ... < y_t$.

Definition 7.5.4. The zerodimensional ideal I is the *syndrome ideal* and its variety $V(I)$ the *syndrome variety*.

Loustaunau and York, in [66], improved the approach introduced by Chen. They suggested to use the FGLM algorithm to make the Groebner computation.
Caboara and Mora, in [13], gave a corrected and optimized version of Chen's algorithm, basing on the studies on the structure of Groebner bases for zerodimensional ideals by Gianni [44] and Kalkbrener [59], who stated Gianni-Kalkbrener theorem (see 3.5.3).

We sketch now the improvements due to M.Sala and E.Orsini.
Consider the syndrome variety $V(I)$ defined by Caboara-Mora in [13] and a correctable syndrome $\mathbf{s} \in (\mathbb{F}_q^m)^{n-k}$; there are some points in the variety that uniquely determine the potential error locations and error values, but, unfortunately, there are also points, called *spurious solutions* (see theorem 7.5.3) from now on, not corresponding directly to some error vector.
Essentially, as explained in [93], the spurious solutions are the points containing zero in correspondence to some error value (the error value cannot be zero) and the ones containing repeated locations (indeed, they must correspond to different positions for the error values). Moreover, they are also the solutions outside the base field.
M.Sala and E.Orsini propose a new syndrome variety eliminating these points.
They consider an $[n, k, d]_q$ cyclic code with $GCD(q, n) = 1$ and give the following

Definition 7.5.5. Let $n \in \mathbb{N}$ be an integer. We denote $p_{ll'} \in \mathbb{F}_q[z_1, ..., z_t]$ as

$$p_{ll'} := \frac{z_l^n - z_{l'}^n}{z_l - z_{l'}},\ 1 \leq l < l' \leq t.$$

The syndrome ideal is $I = (\mathcal{F}_{OS})$ with

$$\mathcal{F}_{OS} = \{f_i, h - j, \chi_i, \lambda_j, p'_{ll'}, 1 \leq l < l' \leq t,\ 1 \leq i \leq n - j,\ j \in S\} \subset \mathbb{F}_q[\mathbf{x}, \mathbf{y}, \mathbf{z}]$$

with

- $f_i := \sum_{l=1}^{t} y_l z_l^j - x_i$
- $h_j := z_j^{n+1} - z_j$;
- $\lambda_j := y_j^{q-1} - 1$;
- $\chi_i := x_i^{q^m} - x_i$;
- $p'_{ll'} = z_{l'} z_l p_{ll'}$

If $\mathcal{Q} := \mathbb{F}_q[x_1, ..., x_{n-k}]$, $\mathcal{G}$ is the usual reduced Groebner basis and for each $\iota = 1, ..., t$, for each l, $\mathcal{G}_\iota := \mathcal{G} \cap \mathcal{Q}[z_t, ..., z_\iota]$, $\mathcal{G}_{\iota l} = \{g \in \mathcal{G}_\iota \setminus \mathcal{G}_{\iota+1},\ deg_\iota(g) = l\}$ and the polynomials are ordered such that their leading terms are ordered w.r.t. lex, then

Theorem 7.5.6. It holds

1. $\mathcal{G} \cap \mathcal{Q}[z_1, ..., z_t] = \bigcup_{i=1}^{t} \mathcal{G}_i$;
2. $\mathcal{G}_i = \bigcup_{\delta=1}^{i} \mathcal{G}_{i\delta}$, $\mathcal{G}_{i\delta} \neq \emptyset$, $1 \leq i \leq t$, $1 \leq \delta \leq i$;
3. $\mathcal{G}_{ii} = \{g_{ii1}\}$, $1 \leq i \leq t$;
4. $\mathsf{T}(g_{ii1}) = z_i^i$, $Lp(g_{ii1}) = 1$;
5. if $1 \leq i \leq t$, $1 \leq \delta \leq i-1$, then $\forall g \in \mathcal{G}_{i\delta}$, and the trailing polynomial is equal to 0.

Let g_{tt1} the unique polynomial in $\mathcal{G}_t$ with $deg_{z_t}(g_{tt1}) = t$:

$$g_{tt1} = z_t^t + \sum_{l=1}^{t} b_{t-l} z_t^{t-l}.$$

T.F.A.E.:

1. there are exactly μ errors;
2. $b_{t-l}(s) = 0$ for $l > \mu$ and $b_{t-\mu}(s) \neq 0$;
3. $g_{tt1}(s, z_t) = z^{t-\mu}(Le(z))$.

This means $\sigma(z) = z^\mu g_{tt1}(s, z^{-1})$, i.e. $g_{tt1} \in \mathcal{Q}[z]$ is a monic polynomial such that

> *given a syndrome vector* $\mathbf{s} \in (\mathbb{F}_{q^m})^{n-k}$*, corresponding to an error of weight* $\mu \leq t$*, its* t *roots are the* μ *location plus zero, counted with multiplicity* $t - \mu$*.*

It is called *general error locator polynomial* of C.

Theorem 7.5.7 ([82])**.** Every cyclic code possesses a general error locator polynomial.

Once we get a general error locator polynomial for C, the decoding algorithm only consists on evaluating it in the syndromes, so its efficiency depends on the sparsity of the involved general error locator polynomial.

Theorem 7.5.8. Let C be a code with error correction capability $t = 1$ and $\mathbf{s}$ a correctable syndrome, then the general error locator polynomial is $\mathcal{L}_C(X, z) = z + a$, $a \in \mathbb{F}_2[X]$. Moreover, there is one error if and only if $a(\mathbf{s}) \neq 0$, being $a(\mathbf{s})$ itself the error location.
Let C be a code with $t = 2$, $\mathbf{s}$ a correctable syndrome and $\overline{z}_1, \overline{z}_2$ the error locations. Then $\mathcal{L}_C(X, z) = z^2 + az + b$, $a, b \in \mathbb{F}_2[X]$ and $b(\mathbf{s}) = \overline{z}_1\overline{z}_2$, $a(\mathbf{s}) = \overline{z}_1 + \overline{z}_2$.
Moreover, there are two errors if and only if $b(\mathbf{s}) \neq 0$, and there is an error if and only if $b(\mathbf{s}) = 0$ and $a(\mathbf{s}) \neq 0$.

We recall here the main theorems stated in [78].

Theorem 7.5.9. Let C a binary $[n, k, d]$ code, with $n \leq 61$ and $d = 3, 4$, $t = 1$. If S is a defining set for C and $\mathcal{L}_C \in \mathbb{F}_q[x_1, .., x_{n-k}][z]$ a general error locator polynomial, four possibilities can occur:

1. if $S = \{m\}$ with $GCD(n, m) = 1$, there exists an integer k mod n such that $\mathcal{L}_C = z + x_1^k$;

2. if $S = \{m, h\}$ with $GCD(h, m) = 1$, there exist two integers m', h' mod n such that $\mathcal{L}_C = z + x_1^{m'}x_2^{h'}$;

3. C is a sub-code of C', of type 1 or 2 and $\mathcal{L}_C = \mathcal{L}_{C'}$;

4. C is equivalent to a code C' of type $1, 2$ or 3 and we can trivially obtain $\mathcal{L}_C$ from $\mathcal{L}_{C'}$

Theorem 7.5.10. Let C be a code with length $n \in \{3, ..., 125\}$, $n \neq 105$ and distance $d = 5, 6$. Then C is equivalent to another code D with $1 \in S_D$.

Theorem 7.5.11. Let C be a binary $[n, k, d]$ code with $n \in \{7, ..., 62\}$, n odd, $d = 5, 6$ and $t = 2$.
Seven possibilities can occur:

1. n is such that C has $S_C = \{0, 1\}$ and $d \leq 5$;

2. C is a BCH code ($S_C = \{1, 3\}$) and $b = x_1^{n-1}(x_1^3 + x_2)$;

3. $S_C = \{1, n-1, l\}$, $l = 0, \frac{n}{3}$ and

$$b = \begin{cases} x_1 x_2^{-1}(1+x_3), \ l = 0 \\ \frac{x_3^3+1}{x_1^{n/3} x_2^{2/3n} x_3 + 1}, \ l = \frac{n}{3}; \end{cases}$$

4. $S_C = \{1, n/l\}$ for some $l \leq 3$;

5. C is one of the following:

$$n = 31, \ S_C = \{1, 15\}; \ n = 31, \ S_C = \{1, 5\}; \ n = 45, \ S_C = \{1, 21\};$$

$$n = 51, \ S_C = \{1, 9\}; \ n = 51, \ S_C = \{0, 1, 5\};$$

6. C is a sub-code of one of those presented above;

7. C is equivalent to one of those presented above.

Even if at present there is no known theoretical proof of the sparsity of general error locator polynomials, there are some experimental evidence, at least in the binary case. Some improvements to the algorithm have been given in [78].
In [83] is stated that

> *Actually*[3] *the number of monomials of* $\mathcal{L}$ *apparently grows linearly, since* $|\mathcal{L}| \leq n$. *We give some theoretical explanations for the sparsity of our polynomials, in all cases except two.*
> *A complete proof for all cases (any and any) seems far beyond our means, at present, but we plan to investigate more and more particular cases, hoping sooner or later to get the profound reason behind the sparsity, whose experimental evidence is apparent (at least in the binary case).*

[3] In the paper [83], $\mathcal{L}$ is the general error locator polynomials

CHAPTER 8

Some experiments on locator polynomials.

8.1 Introduction.

In this chapter we treat some partial results of a joint work with Massimiliano Sala and Teo Mora, connecting the Axis of Evil Theorem to error correcting codes.
In our context, we consider a binary BCH code C of length $n = 2^m - 1$ with error correction capability $t = 2$, correcting simultaneously 1 and 2 errors studying the general error locator polynomial and the related syndrome variety $V(\mathcal{F}_{OS})$ from a different point of view.
Up till now, we have *computed lexicographical reduced Groebner bases of polynomial ideals*. Due to the huge number of variables, such a computation is rather inefficient, so we try to *reverse our point of view*, approaching the problem *à la Moeller*, rather than *à la Buchberger*.
Instead of considering a system of equations, we consider directly the syndrome variety by Orsini and Sala, trying to derive the general error locator polynomial via interpolation.
As explained in the previous chapter, it would be important to prove the *sparsity* of the gen-

eral error locator polynomial.
We will show that Cerlienco-Mureddu Correspondence and the Axis of Evil theorem, with the related algorithm, can be helpful for our purpose.
Indeed, thanks to Cerlienco-Mureddu Correspondence, we can give a precise description of the Groebner escalier associated to $V(\mathcal{F}_{OS})$. Such a description and the properties of the Axis of Evil factorization permit us to reduce appreciably the number of points to deal with.
Moreover, we will see that in some special cases, we can find a structure underlying some sparse general locator polynomials, which involve the cycle structure of the base field and Frobenius automorphism.
Our aim is to prove that the number of terms in the general error locator polynomial grows linearly with the cardinality of the base field.
Since this is still a work in progress, we cannot give here complete results. Anyway, the half-time results we will give in the following sections are rather encouraging.
These partial results have been computed implementing timely procedures, using the programming language provided by Singular [30] and exploiting, as usual, the library pointid.lib by S. Steidel [103] for the Axis of Evil factorization.
Section 8.2 explains out problem in details and gives a precise description of the structure of the Groebner escaliers we have to deal with. Sections 8.3, 8.4 give the first partial results we got in the case of $\mathbb{F}_8, \mathbb{F}_{16}$. Since these results are not optimal, we continued our investigation on $\mathbb{F}_8$, obtaining the results of section 8.5. Finally, in section 8.6, we explain our future projects of generalization for the encouraging results in $\mathbb{F}_8$.

8.2 Our problem.

In this section, we start giving more details about our problem. Precise data for the specific examined cases will be given in the following sections.

Let us consider a binary BCH code C of length $n = 2^m - 1$ for some $m \geq 3$, with error correction capability $t = 2$ and defining set $S_{C'} = \{1, 3\}$.
The complete defining set is $S_C = C_1 \cup C_3$ and we denote by δ the designed distance. We set $d = \delta$ and we have $k = n - |S_C|$. [77].
We deal with the points in the syndrome variety by Sala and Orsini, deciding to correct 1 and 2 errors simultaneously.
More precisely, we start considering all the points of the form

$$(x_1, x_2, z_1, z_2) = (a + b, a^3 + b^3, a, b),$$

where are the variables x_1, x_2 represent the syndromes and z_1, z_2 the locations ($x_1 < x_2 < z_1 < z_2$), letting a, b vary in $\mathbb{F}_q := \mathbb{F}_{2^m}$ *in all possible ways*. The forms assumed by the syndromes come from the ones of polynomials $f_i \in \mathcal{F}_{OS}$: $f_i := \sum_{l=1}^{t} y_l z_l^j - x_i$, where the error values are $y_l = 1$, since we are dealing with a binary code. The related syndromes are therefore $a + b, a^3 + b^3$ (see section 7.5 for more details).
There are q^2 such points, but we have to *exclude the spurious solutions*, not corresponding univocally to an error vector.
We start excluding the point $(0, 0, 0, 0)$, since it corresponds to the absence of errors. Moreover, we exclude the 4-uples of the form $(0, 0, a, a)$, $a \in \mathbb{F}_q \setminus \{0\}$: for $x_1 = x_2 = 0$ we authomatically have the couples of error locations (a, a).
Consequently, the points we have to examine are only the ones of the form $(a+b, a^3+b^3, a, b)$, with $a, b \in \mathbb{F}_q$, $a \neq b$.
Being $a + b, a^3 + b^3$ univocally determined once one knows $a, b \in \mathbb{F}_q$, sometimes, we will identify the 4-tuple $(a + b, a^3 + b^3, a, b)$ with the couple (a, b) and we will write them indifferently.
After the exclusion of spurious solutions, we get $q^2 - q$ distinct points, forming a set we denote by $\mathbf{X}$ and, as usual, we write $I := I(\mathbf{X})$ for the corresponding zerodimensional radical ideal.
We give now a characterization for the lexicographical Groebner escalier ($x_1 < x_2 < z_1 < z_2$) associated to I. In order to describe it, we state the following

Notation 8.2.1. If $\tau \in \mathcal{T}$ is a term and $\mathsf{H} \subset \mathcal{T}$,

$$\tau\mathsf{H} := \{\tau\sigma,\ \sigma \in \mathsf{H}\}.$$

Proposition 8.2.2. With the above notation, set $\mathsf{H} = \{1, x_1, ..., x_1^{q-2}\}$, where q is the cardinality of the base field.
The lexicographical Groebner escalier ($x_1 < x_2 < z_1 < z_2$) of the ideal $I = I(\mathbf{X})$ described as the ideal associated to $\mathbf{X} = \{(a + b, a^3 + b^3, a, b),\ a, b \in \mathbb{F}_{2^m},\ a, \neq b\}$ has the form

$$\mathsf{N}(I) = \mathsf{N}' \cup z_1\mathsf{N}',$$

where

$$\mathsf{N}' = \mathsf{H} \cup x_2\mathsf{H} \cup ... \cup x_2^{\frac{q}{2}-1}\mathsf{H}.$$

Proof: Consider the set $\mathbf{X}$. If we fix $a, b \in \mathbb{F}_{2^m}$ and we consider the associated points

$$P_1 := (a + b, a^3 + b^3, a, b),\ P_2 := (a + b, a^3 + b^3, b, a),$$

clearly P_1, P_2 share the same first two coordinates so, by Cerlienco-Mureddu Correspondence we can partition $\mathbf{X}$ as $\mathbf{X} = \mathbf{X}_1 \sqcup \mathbf{X}_2$, such that if, for some $a, b \in \mathbb{F}_{2^m}$ $(a+b, a^3+b^3, a, b) \in \mathbf{X}_1$,

necessarily $(a+b, a^3+b^3, b, a) \in \mathbf{X}_2$ and if $\mathsf{N}_1 = \mathsf{N}(I(\mathbf{X}_1))$ then $\mathsf{N} = \mathsf{N}(I(\mathbf{X}_1)) \cup z_1\mathsf{N}(I(\mathbf{X}_1))$. We restrict then to $\mathbf{X_1}$.

By hypothesis, $a \neq b \in \mathbb{F}_{2^m}$ hence, clearly, $a+b \neq 0$; on the other hand, $\forall c \in \mathbb{F}_q^*$, $\forall a \in \mathbb{F}_q^*$, $a \neq c$, let $b = c - a$. We have $b \neq a$, $b \neq 0$ and $c = a + b$. Clearly it also holds $c = c + 0$.

The above relations imply that the points in $\mathbf{X}_1$ have $(q-1)$ different first coordinates, so $1, x_1, ..., x_1^{q-2} \in \mathsf{N}$.

Moreover, by the partition formulas of [80], the couples (a, b) such that $a + b = c \in \mathbb{F}_{2^m}^*$ are exactly $\frac{2^m-2}{2}$ if we impose $a, b \neq 0$. Since also $c + 0 = 0$, we add the couple $(c, 0)$, obtaining that *there are* 2^{m-1} *distinct points for each first coordinate*.

The assertion is proved by Cerlienco-Mureddu Correspondence if we can show that, among the points having the first coordinate, it is impossible that two points share also the second coordinate.

What I meant so far, is that if for some $a, b, c, d \in \mathbb{F}_{2^m}^*$ we have

$$\begin{cases} a + b = c + d \neq 0 \\ a^3 + b^3 = c^3 + d^3 \end{cases}$$

then $\{a, b\} = \{c, d\}$.

Indeed, by $a^3 + b^3 = c^3 + d^3$ we have

$$(a+b)(a^2+b^2+ab) = (c+d)(c^2+d^2+cd) \Rightarrow (a+b)^2 + ab = (c+d)^2 + cd \Rightarrow ab = cd.$$

The elements a, b, c, d are then the roots in $\mathbb{F}_{2^m}$ of $x^2 + (a+b)x + ab$. Being them only two [65] and since a, b are obviously roots of the trinomial, we necessarily have $\{a, b\} = \{c, d\}$ and we can conclude. $\diamond$

Since the Groebner escalier has always this shape, we know that z_1^2, z_2 always belongs to the monomial basis $\mathsf{G}(I)$.

Moreover, we know that $z_2 = z_1 + x_1 \in I$, since for each couple of elements $a, b \in \mathbb{F}_q$, $(a+b) + a = b$ (actually it even belongs to the lexicographical reduced Groebner basis of I, being $x_1, z_1 \in \mathsf{N}(I)$), so, once one has determined a, b can be simply computed via that linear (and very sparse) relation.

This implies that, among the polynomials in a minimal lexicographical Groebner basis of I, we only have to deal with the one whose leading term is z_1^2, which allows to compute the values for the first error.

By the evident symmetry of $\mathsf{N}(I)$, applying the Axis of Evil algorithm to the points (x_1, x_2, z_1, z_2), for the factorization of the required polynomial, we get two factors $F_a := z_1 + f_a(x_1, x_2)$ and $F_b := z_1 + f_b(x_1, x_2)$.

Moreover $\mathbf{X}$ is partitioned in two subsets $\mathbf{Z}_a, \mathbf{Z}_b \subset \mathbf{X}$, with $|\mathbf{Z}_a| = |\mathbf{Z}_b| = \frac{1}{2}|\mathbf{X}| = \binom{q}{2}$[1] such that:

- F_a vanishes on the points of $\mathbf{Z}_a$
- F_b vanishes on the points of $\mathbf{Z}_b$
- $(x_1, x_2, z_1, z_2) \in \mathbf{Z}_a \Leftrightarrow (x_1, x_2, z_2, z_1) \in \mathbf{Z}_b$.

Then, we can restrict to one of the subsets, say $\mathbf{Z}_a$ and compute F_a: the other points come from $z_2 = z_1 + x_1$.
Therefore, we arrange the points in couples of the form

$$[(a+b, a^3+b^3, a, b), (a+b, a^3+b^3, b, a)],$$

according to their first three coordinates, since we do not need any computation involving z_2.
Then, we choose one point for each couple[2]. The choice of the points influences the sparsity of the locator polynomial F_a. Our aim is to determine locator polynomials linearly growing with the cardinality of the base field, characterizing them, if possible, with a pattern, in order to generalize the construction to larger cases.
We start reporting here the partial results obtained for $\mathbb{F}_8, \mathbb{F}_{16}, \mathbb{F}_{32}$.

8.3 The case of $\mathbb{F}_8$: cyclic configurations.

The simplest base field for our study is the one corresponding to $m = 3$, namely

$$\mathbb{F}_8 = \{0, 1, a, a+1, a^2, a^2+1, a^2+a, a^2+a+1\},$$

with primitive element a: $a^3 = a + 1$.

For brevity's sake, from now on, we will set $\mathbb{F}_8 = \{0, 1, a, a^2, a^3, a^4, a^5, a^6\}$.
Our code is a binary $[n, k, d]$ BCH code with $n = 2^3 - 1 = 7$, $d = \delta = 7$, $k = 1$. Its error correction capability is $t = 2$ and we suppose to correct 1 and 2 errors simultaneously.
As explained in section 8.2, the points we first take in to account are $64 = 8^2$, and they have the form

$$(a+b, a^3+b^3, a, b),\ a, b \in \mathbb{F}_8,$$

[1] They mirror the simmetry of the Groebner escalier.
[2] This fact represents a further proof of $\frac{1}{2}|\mathbf{X}| = \binom{q}{2}$, since the idea is that we are taking the couples $(a, b) \in (\mathbb{F}_q)^2$, $a \neq b$ disregarding the entries' order.

where $x_1 = a + b$, $x_2 = a^3 + b^3$ are the syndromes and $z_1 = a$, $z_2 = b$ the error locations.
We discard now the spurious solutions, namely the points of the form $(0, 0, a, a)$: if $x_1 = x_2 = 0$ we have the 8 couples of locations (a, a).
Applying the Axis of Evil algorithm on the remaining 56 points (x_1, x_2, z_1, z_2), we get two polynomials $F_a := z_1 + f_a(x_1, x_2)$ e $F_b := z_1 + f_b(x_1, x_2)$ and a partition of the 56 points in two subsets $\mathbf{Z}_a, \mathbf{Z}_b$ of cardinality 28, satisfying the properties stated in the previous section.
We arrange then the 56 points in 28 couples, according to their first three coordinates, i.e. each couple will be of the form

$$[(a + b, a^3 + b^3, a, b), (a + b, a^3 + b^3, b, a)].$$

More precisely, we get a list P containing the following 28 couples:

$$[(a, a^3, 0, a), (a, a^3, a, 0)],$$
$$[(a^2, a^6, 0, a^2), (a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, 0, a^3), (a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, 0, a^4), (a^4, a^5, a^4, 0)],$$
$$[(a^5, a, 0, a^5), (a^5, a, a^5, 0)],$$
$$[(a^6, a^4, 0, a^6), (a^6, a^4, a^6, 0)],$$
$$[(1, 1, 0, 1), (1, 1, 1, 0)],$$
$$[(a^4, a^4, a, a^2), (a^4, a^4, a^2, a)],$$
$$[(1, a^5, a, a^3), (1, a^5, a^3, a)],$$
$$[(a^2, a^2, a, a^4), (a^2, a^2, a^4, a)],$$
$$[(a^6, 1, a, a^5), (a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a, a^6), (a^5, a^6, a^6, a)],$$
$$[(a^3, a, a, 1), (a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3), (a^5, 1, a^3, a^2)],$$
$$[(a, a, a^2, a^4), (a, a, a^4, a^2)],$$
$$[(a^3, a^5, a^2, a^5), (a^3, a^5, a^5, a^2)],$$
$$[(1, a^3, a^2, a^6), (1, a^3, a^6, a^2)],$$

$$[(a^6, a^2, a^2, 1), (a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4), (a^6, a^3, a^4, a^3)],$$
$$[(a^2, a^4, a^3, a^5), (a^2, a^4, a^5, a^3)],$$
$$[(a^4, a, a^3, a^6), (a^4, a, a^6, a^3)],$$
$$[(a, a^6, a^3, 1), (a, a^6, 1, a^3)],$$
$$[(1, a^6, a^4, a^5), (1, a^6, a^5, a^4)],$$
$$[(a^3, 1, a^4, a^6), (a^3, 1, a^6, a^4)],$$
$$[(a^5, a^4, a^4, 1), (a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^5, a^6), (a, a^2, a^6, a^5)],$$
$$[(a^4, a^3, a^5, 1), (a^4, a^3, 1, a^5)],$$
$$[(a^2, a^5, a^6, 1), (a^2, a^5, 1, a^6)].$$

Thanks to proposition 8.2.2, the tower structure of the Groebner escalier we have to work with is

x_2^3	$x_1x_2^3$	$x_1^2x_2^3$	$x_1^3x_2^3$	$x_1^4x_2^3$	$x_1^5x_2^3$	$x_1^6x_2^3$
x_2^2	$x_1x_2^2$	$x_1^2x_2^2$	$x_1^3x_2^2$	$x_1^4x_2^2$	$x_1^5x_2^2$	$x_1^6x_2^2$
x_2	x_1x_2	$x_1^2x_2$	$x_1^3x_2$	$x_1^4x_2$	$x_1^5x_2$	$x_1^6x_2$
1	x_1	x_1^2	x_1^3	x_1^4	x_1^5	x_1^6

In order to deal with this problem, we employ the original Axis of Evil algorithm. Indeed, as explained in chapter 3 even if the minimal Groebner basis we get is not reduced, the linear factors produced are. Moreover, the interpolation step in algorithm 5, line 19 ensures that the maximal number of terms composing each linear factor is $|\mathbf{X}| + 1 = |\mathsf{N}(I(\mathbf{X}))| + 1$. We compute the polynomials using Singular. More precisely, we run on the points the facGBIdeal procedure from the library `pointid.lib` [30, 103].
For example, for the following choice of the points[3]

$$list\, Q1 = P[1][1], P[2][2], P[3][2], P[4][2], P[5][1], P[6][1], P[7][1], P[15][2], P[10][2],$$
$$P[13][1], P[8][1], P[12][1], P[11][1], P[9][2], P[22][1], P[20][1], P[16][1], P[21][2],$$
$$P[14][2], P[18][2], P[17][1], P[26][1], P[28][2], P[24][2], P[27][2], P[25][2], P[19][1], P[23][2];$$

[3]The configuration list is
$[1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2]$.

we get

$$list\,U1 = \text{facGBIdeal}(Q1);$$
$$poly\,\text{tcontrollo1} = U1[2][3][1];$$
$$ncols(\text{coef}(\text{controllo1}, x_1x_2z_1));$$
$$\Rightarrow z_1 + x_1^6x_2^3 + ax_1^5x_2^3 + a^2x_1^4x_2^3 + a^3x_1^3x_2^3 + a^4x_1^2x_2^3 + a^5x_1x_2^3 + a^6x_2^3 + ax_1^6x_2^2 + x_1^5x_2^2 +$$
$$ax_1^4x_2^2 + a^5x_1^3x_2^2 + a^2x_1^2x_2^2 + a^4x_1x_2^2 + a^5x_2^2 + x_1^6x_2 + a^4x_1^5x_2 + a^3x_1^4x_2 + a^4x_1^3x_2 +$$
$$a^5x_1^2x_2 + a^6x_1x_2 + a^2x_2 + a^2x_1^6 + a^3x_1^4 + a^6x_1^3 + x_1^2 + ax_1 + a^4$$

As a first result, we found 7 configurations presenting an easy structure, leading to polynomials made up of 18 terms.
It is possible to describe such a structure in a very precise way.
The 7 configurations are connected to *cyclic permutations of powers of the primitive element* a in the sense described below[4].
We choose the first points configuration so that we get:

- 7 points whose third coordinate is a^{i_1}, $i_1 \in \{1, ..., 7\}$;
- 6 points whose third coordinate is a^{i_2}, $i_2 \in \{1, ..., 7\} \setminus \{i_1\}$;
- ...
- 1 point points whose third coordinate is a^{i_7}, $i_7 \in \{1, ..., 7\} \setminus \{i_1, ..., 1_6\}$;
- no points whose third coordinate is 0.

We summarize such a choice in a table

Number of points	Third coordinate
7	i_1
6	i_2
5	i_3
4	i_4
3	i_5
2	i_6
1	i_7

Then, for the same values of $i_1, ..., i_7$, which are pairwise different by construction, we choose another point configuration such that

[4]They are exactly 7 since the powers of a are 7 and the cyclic permutations of a cycle of length h are h.

- 7 points whose third coordinate is a^{i_7};
- 6 points whose third coordinate is a^{i_1};
- ...
- 1 point points whose third coordinate is a^{i_6};
- no points points whose third coordinate is 0.

This choice can be summarized in an analogous table and we can proceed this way for each cyclic permutation of $i_1, ..., i_7$, obtaining another configuration in correspondence.
We point out that each cyclic permutation corresponds to *only one* point configuration, by the structure of the couples in P.
Indeed, let us consider, for example the cyclic permutation associated to the table above.
Among the 56 non spurious points, only 7 of them have i_1 as third coordinate: they are the ones of the form $(i_1 + b, i_1^3 + b^3, i_1, b)$, with $b \neq i_1 \in \mathbb{F}_8$: since $|\mathbb{F}_8| = 8$ and since we exclude the case $(0, 0, i_1, i_1)$, there are exactly 7 possible values for b. Thus, we have no choice and we have to take exactly that points.
Then we consider the second row of the table: we need to choose 6 points with $i_2 \neq i_1$ as third coordinate. Again there are 7 points of this shape.
Nevertheless, among these 7 points, there is also $(i_2+i_1, i_2^3+i_1^3, i_2, i_1)$. Since for our configuration we have choosen *all* the 7 points with third coordinate equal to i_1, $(i_1+i_2, i_1^3+i_2^3, i_1, i_2)$ belongs to our configuration.
On the other hand, for each couple of the form

$$[(a + b, a^3 + b^3, a, b), (a + b, a^3 + b^3, b, a)], a, b \in \mathbb{F}_8, a \neq b,$$

we want to choose *only one point*, so *we cannot choose* $(i_2 + i_1, i_2^3 + i_1^3, i_2, i_1)$ and this implies that we have only 6 points with i_2 as third coordinate to take into account and again we have no choice while picking them. Following the same line for $i_3, ..., i_7$ we get the *unique* point configuration associated to the table above.
Considering a cyclic permutation of $i_1, .., i_7$ in the sense explained before we get another (unique) point configuration.
Every polynomial we obtain by applying algorithm 5 to the unique point configuration associated to one of the 7 cyclic permutation of $i_1, ..., i_7$ is composed by the same number of terms, i.e. 18.
Let us examine two of them. The precise data for all the 7 points configurations are contained in appendix B, B.1.1.

Configuration 1

This configuration corresponds to the permutation represented in following table:

Number of points	Third coordinate
7	a^2
6	a^3
5	a^4
4	a^5
3	a^6
2	1
1	a

The unique point configuration associated to that table is:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a^2, a)],$$
$$[(1, a^5, a^3, a)],$$
$$[(a^2, a^2, a^4, a)],$$
$$[(a^6, 1, a^5, a)],$$

$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$
$$[(a^3, a^5, a^2, a^5)],$$
$$[(1, a^3, a^2, a^6)],$$
$$[(a^6, a^2, a^2, 1)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, a^3, 1)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, a^4, 1)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

If we draw the tower structure of these points, disregarding the fourth coordinate as a consequence of proposition 8.2.2, we get

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^4	a^6,a^3,a^3	1,a^6,a^4
a,a^6,a^3	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,a^2	1,a^3,a^2
a,a,a^2	a^2,a^2,a^4	a^3,a,1	a^4,a^4,a^2	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a^3
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

while the configuration list is

$$[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

and the associated locator polynomial turns out to be

$$z_1 + a^6x_1^5x_2^3 + a^5x_1^4x_2^3 + a^5x_1^3x_2^3 + a^3x_1^2x_2^3 + a^6x_1x_2^3 + a^3x_2^3 + a^5x_1^6x_2^2 +$$
$$a^3x_1^5x_2^2 + a^6x_1^4x_2^2 + a^3x_1^3x_2^2 + a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^5x_1^3x_2 + a^3x_1x_2 +$$
$$a^3x_1^4 + ax_1 + a^6$$

and it is made up of 18 terms.

Configuration 2

This configuration corresponds to the cyclic permutation summarized in the table below:

Number of points	Third coordinate
7	a^3
6	a^4
5	a^5
4	a^6
3	1
2	a
1	a^2

whose associated configuration list is

$$[2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1].$$

Again it leads to a polynomial made up of 18 terms (see appendix B).

Remark 8.3.1. We remark that the cyclic permutations are fundamental for getting 18 terms. If we break the pattern even by *one only point*, we get a remarkably denser polynomial. For example, if we modify a little configuration 1, *changing only the last point*, we get

Configuration 7.2

Number of points	Third coordinate
7	a^2
6	a^3
5	a^4
4	a^5
3	a^6
2	1
1	0

and we obtain a general error locator polynomial made up of 25 terms (see appendix B, B.1.1 for further details).

We notice that the "cyclic permutations" we are considering arise from the multiplication by the primitive element a. For example, in configuration 1 we have 7 points whose third coordinate is a^2, while in configuration 2 we have 7 points whose third coordinate is a^3.

It is easy to verify that this happens for all the entries in the tables associated to configurations $1, 2$ and that it happens also for configurations $3, ..., 7$, whose data are displayed in appendix B,B.1.2.
In order to study the similarities among the polynomials we got, we first consider the 7×4 matrix

$$M = \begin{bmatrix} x_1^6 & x_1^6x_2 & x_1^6x_2^2 & x_1^6x_2^3 \\ x_1^5 & x_1^5x_2 & x_1^5x_2^2 & x_1^5x_2^3 \\ x_1^4 & x_1^4x_2 & x_1^4x_2^2 & x_1^4x_2^3 \\ x_1^3 & x_1^3x_2 & x_1^3x_2^2 & x_1^3x_2^3 \\ x_1^2 & x_1^2x_2 & x_1^2x_2^2 & x_1^2x_2^3 \\ x_1 & x_1x_2 & x_1x_2^2 & x_1x_2^3 \\ 1 & x_2 & x_2^2 & x_2^3 \end{bmatrix} \tag{8.1}$$

Then, we describe the coefficients of the polynomial p_i associated to configuration i, $i = 1, ..., 7$ with a 7×4 matrix $A^{[i]} = (a^{[i]}_{l,m})$, such that $a^{[i]}_{l,m}$ is the coefficient of the term $m_{l,m}$ in p_i.
We list here only the matrices $A^{[1]}, A^{[2]}$, associated to configurations $1, 2$. The reader can find the other ones in appendix B, B.1.2.

Configuration 1**:**
corresponds to

$$A^{[1]} = \begin{pmatrix} 0 & 0 & a^5 & 0 \\ 0 & a^4 & a^3 & a^6 \\ a^3 & 0 & a^6 & a^5 \\ 0 & a^5 & a^3 & a^5 \\ 0 & 0 & a^6 & a^3 \\ a & a^3 & 0 & a^6 \\ a^6 & 0 & 0 & a^3 \end{pmatrix}$$

Configuration 2
corresponds to

$$A^{[2]} = \begin{pmatrix} 0 & 0 & a & 0 \\ 0 & a^4 & 1 & 1 \\ 1 & 0 & a^4 & 1 \\ 0 & 1 & a^2 & a \\ 0 & 0 & a^6 & 1 \\ a & 1 & 0 & a^4 \\ 1 & 0 & 0 & a^2 \end{pmatrix}$$

The coefficients summarized above present some common properties.
In general, the matrices $A^{[1]}, ..., A^{[7]}$ have a very precise shape, hightlighted in the following "general" matrix:

$$A^{[\text{gen}]} = \begin{pmatrix} 0 & 0 & C & 0 \\ 0 & a^4 & D & A \\ D & 0 & E & B \\ 0 & B & F & C \\ 0 & 0 & a^6 & D \\ a & D & 0 & E \\ A & 0 & 0 & F \end{pmatrix} ; \ A, B, C, D, E, F \in \mathbb{F}_8.$$

The existence of an $A^{[\text{gen}]}$, whose entries summarize the coefficients of the polynomials $p_1, .., p_7$ of the 7 configurations obtained by the cyclic permutations, tells us that the multiplication by a we perform to swich from a configuration to another one "preserves the supports of polynomials", in the sense that

$$\text{Supp}(p_1) = \text{Supp}(p_2) = ... = \text{Supp}(p_7).$$

Moreover, as we can see in the above $A^{[gen]}$, some values are *stable* among $A^{[1]}, ..., A^{[7]}$, namely $a^{[gen]}_{2,2} = a^4$, $a^{[gen]}_{5,3} = a^6$, $a^{[gen]}_{6,1} = a$.

Notice that the capital letters appearing in the table (i.e. the different non-stable values for the coefficients) are 6, i.e. $|\mathbb{F}_8^*| - 1$.
We can get general formulas for the values $A, B, C, D, E, F \in \mathbb{F}_8$.
Each configuration is identified by the number of points $(\mathit{a}+\mathit{b}, \mathit{a}^3+\mathit{b}^3, \mathit{a}, \mathit{b})$ for each appearing third coordinate , i.e. the number of occurrences of some a as third coordinate.
For each configuration, we denote by M the value of the third coordinate a appearing *once*[5]

[5]For example, for configuration 1, we have $M = a$ and for configuration 2, $M = a^2$.

and we get

$$\begin{aligned} A &= a^5 M \\ B &= a^3 M^2 \\ C &= a^2 M^3 \\ D &= a^6 M^4 \\ E &= a M^5 \\ F &= a^4 M^6 \end{aligned} \tag{8.2}$$

If, instead, we denote by M the value of the third coordinate *a* appearing *twice*, we get the set of formulas:

$$\begin{aligned} A &= a^6 M \\ B &= a^5 M^2 \\ C &= a^5 M^3 \\ D &= a^3 M^4 \\ E &= a^6 M^5 \\ F &= a^3 M^6 \end{aligned} \tag{8.3}$$

For 8.2 we have a sort of "symmetry", since we have $a^5M \rightarrow aM^5$, $a^3M^2 \rightarrow a^2M^3$, $a^6M^4 \rightarrow a^4M^6$, which is not mirrored in 8.3.
Choosing M as the value of the third coordinate appearing 7 times, 6 times and so on, we get different formulas. More precisely the powers of M do not change, but the multiplicative coefficients vary. The entire set of formulas is displayed in appendix B.
The set of formulas 8.2 is connected to the structure of cycles in $\mathbb{F}_8$ and the same happens for the all the other ones (appendix B, B.1.2).
This means that the multiplication by a, i.e. the transformation among the 7 points configurations we have, preserves the cycles in $\mathbb{F}_8$.
We recall that the cycles of $\mathbb{F}_8$ are:
α) $\mathbf{a} \rightarrow a^2 \rightarrow a^4 \rightarrow \mathbf{a}$;
β) $\mathbf{a^3} \rightarrow a^6 \rightarrow a^5 \rightarrow \mathbf{a^3}$;
γ) $\mathbf{a^7} = 1$.
δ) $\mathbf{0}$.
We consider the elements having the minimal exponent of a in each cycle as preferential

representatives of the corresponding cycle[6].
Consider the set of formulas 8.2. The powers $(1, 2, 4)$ of M (which are exactly the exponents of the cycle α), are multiplied to powers of a corresponding to cycle β. The powers $(3, 6, 5)$ of M are multiplied to powers of a corresponding to cycle α, so we can summarize the obtained relations as (powers of M, powers of a), namely: $(\alpha, \beta), (\beta, \alpha)$.
We have similar relations for the other sets of formulas. For example, for 8.3 we have $(\alpha, \beta), (\beta, \beta)$.

8.4 The case of $\mathbb{F}_{16}$: cyclic configurations.

Drove by the simple structure of the configurations described in the previous section, we try to generalize them, enlarging the base field.
Consider then

$$\mathbb{F}_{16} = \{0, a, a^2, a^3, a^4, a^5, a^6, a^7, a^8, a^9, a^{10}, a^{11}, a^{12}, a^{13}, a^{14}, 1\},$$

with minimal polynomial $a^4 + a + 1$[7].
Suppose to have again a binary $[n, k, d]$ BCH code C of length $n = 15$, with error correction capability $t = 2$, designed distance $\delta = 5$, distance $d = \delta = 5$. We have $k = 7$ and we suppose again to correct 1 and 2 errors simultaneously.
At the beginning we have $16^2 = 256$ points but, excluding as for $\mathbb{F}_8$ the spurious ones, corresponding to couples of the form $(z_1, z_2) = (a, a)$, we get 240 points we arrange into a list P of 120 couples.
By the structure of the Groebner escalier 8.2.2, we take again one point for each couple, conveniently chosen w.r.t. the third coordinate.
The cyclic configurations of 8.3 can be easily generalized to the case of $\mathbb{F}_{16}$, and we have concretely produced them. They are 15, namely as many as the number of cyclic permutations of the elements in $\mathbb{F}_{16}^*$.
We show again some of them, referring to appendix B, B.2.1, for details.
The first configuration we could find is
Configuration 1:

[6] They are the elements in boldface font.
[7] We follow again the representation of the elements in the field as powers of the primitive element.

Number of points	Third coordinate
15	a
14	a^2
13	a^3
12	a^4
11	a^5
10	a^6
9	a^7
8	a^8
7	a^9
6	a^{10}
5	a^{11}
4	a^{12}
3	a^{13}
2	a^{14}
1	1

The configuration list is

$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$

and the associated locator polynomial is made up of 85 terms

$$z_1 + a^{14}x_1^{14}x_2^7 + ax_1^{13}x_2^7 + a^7x_1^{12}x_2^7 + a^{11}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{12}x_1^9x_2^7 + a^9x_1^8x_2^7 + a^4x_1^7x_2^7+$$
$$a^3x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^8x_1^4x_2^7 + a^{13}x_1^3x_2^7 + a^6x_1^2x_2^7 + a^2x_1x_2^7 + a^5x_2^7 + a^{11}x_1^{14}x_2^6+$$
$$a^5x_1^{13}x_2^6 + a^4x_1^{10}x_2^6 + a^3x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^8x_1^7x_2^6 + a^{13}x_1^6x_2^6 + a^6x_1^5x_2^6 + a^2x_1^4x_2^6+$$
$$a^5x_1^3x_2^6 + a^{14}x_1^2x_2^6 + ax_1x_2^6 + a^7x_2^6 + a^4x_1^{13}x_2^5 + a^{10}x_1^{11}x_2^5 + a^8x_1^{10}x_2^5 + a^{13}x_1^9x_2^5 + a^6x_1^8x_2^5+$$
$$a^2x_1^7x_2^5 + a^5x_1^6x_2^5 + a^{14}x_1^5x_2^5 + ax_1^4x_2^5 + a^7x_1^3x_2^5 + a^{11}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{12}x_2^5 + a^{13}x_1^{12}x_2^4+$$
$$a^6x_1^{11}x_2^4 + a^2x_1^{10}x_2^4 + a^5x_1^9x_2^4 + a^{14}x_1^8x_2^4 + ax_1^7x_2^4 + a^7x_1^6x_2^4 + a^{11}x_1^5x_2^4 + a^9x_1^4x_2^4+$$
$$a^9x_1^2x_2^4 + a^3x_2^4 + a^6x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^3x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^3x_1^6x_2^2 + a^{10}x_1^5x_2^2+$$
$$a^8x_1^4x_2^2 + a^6x_1^2x_2^2 + a^2x_1x_2^2 + a^5x_2^2 + a^{11}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{12}x_1^{12}x_2 + a^9x_1^{11}x_2+$$
$$a^4x_1^{10}x_2 + a^{10}x_1^8x_2 + a^{13}x_1^6x_2 + a^6x_1^5x_2 + a^2x_1^4x_2 + a^{14}x_1^2x_2 + a^7x_2 + a^9x_1^{14} + a^3x_1^{12} + a^{10}x_1^{11}+$$
$$a^{13}x_1^9 + a^6x_1^8 + a^5x_1^6 + a^{14}x_1^5 + a^7x_1^3 + a^{11}x_1^2 + a^{10}x_1 + a^{12}$$

If, in analogy with 8.3, we "cyclically permute" the configuration, i.e. we multiply by the primitive element a, we get fourteen other configurations, leading to fourteen locator poly-

nomials, each one made up of 85 terms. Let us see another configuration.
Configuration 2:

Number of points	Third coordinate
15	a^2
14	a^3
13	a^4
12	a^5
11	a^6
10	a^7
9	a^8
8	a^9
7	a^{10}
6	a^{11}
5	a^{12}
4	a^{13}
3	a^{14}
2	1
1	a

The configuration list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,$$
$$1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,$$
$$1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1$$

and the locator is again made up of 85 terms (see appendix B, section B.2).

As for the cyclic configurations in $\mathbb{F}_8$ we consider the 15×8 matrix

$$M = \begin{bmatrix} x_1^{14} & x_1^{14}x_2 & x_1^{14}x_2^2 & x_1^{14}x_2^3 & x_1^{14}x_2^4 & x_1^{14}x_2^5 & x_1^{14}x_2^6 & x_1^{14}x_2^8 \\ x_1^{13} & x_1^{13}x_2 & x_1^{13}x_2^2 & x_1^{13}x_2^3 & x_1^{13}x_2^4 & x_1^{13}x_2^5 & x_1^{13}x_2^6 & x_1^{13}x_2^8 \\ x_1^{12} & x_1^{12}x_2 & x_1^{12}x_2^2 & x_1^{12}x_2^3 & x_1^{12}x_2^4 & x_1^{12}x_2^5 & x_1^{12}x_2^6 & x_1^{12}x_2^8 \\ x_1^{11} & x_1^{11}x_2 & x_1^{11}x_2^2 & x_1^{11}x_2^3 & x_1^{11}x_2^4 & x_1^{11}x_2^5 & x_1^{11}x_2^6 & x_1^{11}x_2^8 \\ x_1^{10} & x_1^{10}x_2 & x_1^{10}x_2^2 & x_1^{10}x_2^3 & x_1^{10}x_2^4 & x_1^{10}x_2^5 & x_1^{10}x_2^6 & x_1^{10}x_2^8 \\ x_1^9 & x_1^9x_2 & x_1^9x_2^2 & x_1^9x_2^3 & x_1^9x_2^4 & x_1^9x_2^5 & x_1^9x_2^6 & x_1^9x_2^8 \\ x_1^8 & x_1^8x_2 & x_1^8x_2^2 & x_1^8x_2^3 & x_1^8x_2^4 & x_1^8x_2^5 & x_1^8x_2^6 & x_1^8x_2^8 \\ x_1^7 & x_1^7x_2 & x_1^7x_2^2 & x_1^7x_2^3 & x_1^7x_2^4 & x_1^7x_2^5 & x_1^7x_2^6 & x_1^7x_2^8 \\ x_1^6 & x_1^6x_2 & x_1^6x_2^2 & x_1^6x_2^3 & x_1^6x_2^4 & x_1^6x_2^5 & x_1^6x_2^6 & x_1^6x_2^8 \\ x_1^5 & x_1^5x_2 & x_1^5x_2^2 & x_1^5x_2^3 & x_1^5x_2^4 & x_1^5x_2^5 & x_1^5x_2^6 & x_1^5x_2^8 \\ x_1^4 & x_1^4x_2 & x_1^4x_2^2 & x_1^4x_2^3 & x_1^4x_2^4 & x_1^4x_2^5 & x_1^4x_2^6 & x_1^4x_2^8 \\ x_1^3 & x_1^3x_2 & x_1^3x_2^2 & x_1^3x_2^3 & x_1^3x_2^4 & x_1^3x_2^5 & x_1^3x_2^6 & x_1^3x_2^8 \\ x_1^2 & x_1^2x_2 & x_1^2x_2^2 & x_1^2x_2^3 & x_1^2x_2^4 & x_1^2x_2^5 & x_1^2x_2^6 & x_1^2x_2^8 \\ x_1 & x_1x_2 & x_1x_2^2 & x_1x_2^3 & x_1x_2^4 & x_1x_2^5 & x_1x_2^6 & x_1x_2^8 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 & x_2^8 \end{bmatrix} \tag{8.4}$$

and we use similar matrices in order to summarize the coefficients of the polynomials associated to our configurations (the whole list is in B.2.2).

Configuration 1

$$A^{[1]} = \begin{pmatrix} a^9 & a^{11} & 0 & a^6 & 0 & 0 & a^{11} & a^{14} \\ 0 & a^{12} & 0 & 0 & 0 & a^4 & a^5 & a \\ a^3 & a^{12} & 0 & 0 & a^{13} & 0 & 0 & a^7 \\ a^{10} & a^9 & 0 & 0 & a^6 & a^{10} & 0 & a^{11} \\ 0 & a^4 & a^2 & 0 & a^2 & a^8 & a^4 & 1 \\ a^{13} & 0 & 0 & 0 & a^5 & a^{13} & a^3 & a^{12} \\ a^6 & a^{10} & 0 & 0 & a^{14} & a^6 & a^{10} & a^9 \\ 0 & 0 & 0 & a^{10} & a & a^2 & a^8 & a^4 \\ a^5 & a^{13} & a^3 & 0 & a^7 & a^5 & a^{13} & a^3 \\ a^{14} & a^6 & a^{10} & 0 & a^{11} & a^{14} & a^6 & a^{10} \\ 0 & a^2 & a^8 & 0 & a^9 & a & a^2 & a^8 \\ a^7 & 0 & 0 & a^3 & 0 & a^7 & a^5 & a^{13} \\ a^{11} & a^{14} & a^6 & 0 & a^9 & a^{11} & a^{14} & a^6 \\ a^{10} & 0 & a^2 & 0 & 0 & a^{10} & a & a^2 \\ a^{12} & a^7 & a^5 & 0 & a^3 & a^{12} & a^7 & a^5 \end{pmatrix}$$

Configuration 2

$$A^{[2]} = \begin{pmatrix} a^{11} & a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^{10} \\ 0 & a^{12} & 0 & 0 & 0 & a^{7} & a^{5} & a^{13} \\ a^{7} & a^{13} & 0 & 0 & a^{5} & 0 & 0 & a^{5} \\ 1 & a^{11} & 0 & 0 & a^{14} & 1 & 0 & a^{10} \\ 0 & a^{7} & a^{2} & 0 & a^{11} & a^{14} & a^{7} & 1 \\ a^{5} & 0 & 0 & 0 & 1 & a^{5} & a^{7} & a^{13} \\ a^{14} & 1 & 0 & 0 & a^{10} & a^{14} & 1 & a^{11} \\ 0 & 0 & 0 & a^{10} & a^{13} & a^{11} & a^{14} & a^{7} \\ 1 & a^{5} & a^{7} & 0 & a^{5} & 1 & a^{5} & a^{7} \\ a^{10} & a^{14} & 1 & 0 & a^{10} & a^{10} & a^{14} & 1 \\ 0 & a^{11} & a^{14} & 0 & a^{9} & a^{13} & a^{11} & a^{14} \\ a^{5} & 0 & 0 & a^{7} & 0 & a^{5} & 1 & a^{5} \\ a^{10} & a^{10} & a^{14} & 0 & a^{11} & a^{10} & a^{10} & a^{14} \\ a^{10} & 0 & a^{11} & 0 & 0 & a^{10} & a^{13} & a^{11} \\ a^{13} & a^{5} & 1 & 0 & a^{7} & a^{13} & a^{5} & 1 \end{pmatrix}$$

As in the case of $\mathbb{F}_8$, one can find a general matrix, summarizing the reciprocal relations among the coefficients of each locator polynomial:

$$A^{[\text{gen}]} = \begin{pmatrix} B & A & 0 & C & 0 & 0 & A & D \\ 0 & a^{12} & 0 & 0 & 0 & E & a^{5} & F \\ G & H & 0 & 0 & I & 0 & 0 & L \\ M & B & 0 & 0 & C & M & 0 & A \\ 0 & E & a^{2} & 0 & N & O & E & 1 \\ I & 0 & 0 & 0 & P & I & G & H \\ C & M & 0 & 0 & D & C & M & B \\ 0 & 0 & 0 & a^{10} & F & N & O & E \\ P & I & G & 0 & L & P & I & G \\ D & C & M & 0 & A & D & C & M \\ 0 & N & O & 0 & a^{9} & F & N & O \\ L & 0 & 0 & G & 0 & L & P & I \\ A & D & C & 0 & B & A & D & C \\ a^{10} & 0 & N & 0 & 0 & a^{10} & F & N \\ H & L & P & 0 & G & H & L & P \end{pmatrix} ; A, B, C, ..., P \in \mathbb{F}_{16}.$$

Multiplication by a again preserves the support of the locator polynomials and, again, some values are stable among $A^{[1]}, ..., A^{[15]}$.

Notice that the capital letters appearing in the table (the different non-stable values for the coefficients) are 14, i.e. again one less than the number of configurations.
We can find formulas for the letters $A - P$, depending on a value Q. If Q is the value of the first coordinate appearing once, one gets

$$A = a^{12}Q^{14}$$
$$B = a^{7}Q^{2}$$
$$C = a^{13}Q^{8}$$
$$D = a^{3}Q^{11}$$
$$E = aQ^{3}$$
$$F = a^{4}Q^{12}$$
$$G = a^{14}Q^{4}$$
$$H = a^{11}Q$$
$$I = a^{6}Q^{7}$$
$$L = a^{9}Q^{13}$$
$$M = a^{5}Q^{5}$$
$$N = a^{8}Q^{9}$$
$$O = a^{2}Q^{6}$$
$$P = a^{10}Q^{10}$$

If, instead, Q is the value of the first coordinate appearing twice, one gets

$$A = a^{11}Q^{14}$$
$$B = a^{9}Q^{2}$$
$$C = a^{6}Q^{8}$$
$$D = a^{14}Q^{11}$$
$$E = a^{4}Q^{3}$$
$$F = aQ^{12}$$
$$G = a^{3}Q^{4}$$

$$H = a^{12}Q$$
$$I = a^{13}Q^7$$
$$L = a^7Q^{13}$$
$$M = a^{10}Q^5$$
$$N = a^2Q^9$$
$$O = a^8Q^6$$
$$P = a^5Q^{10}$$

As for $\mathbb{F}_8$, if we choose differently Q, the formulas change only on the multiplicative coefficient of Q, not in the power.
The cycles in $\mathbb{F}_{16}$ are

α') $\mathbf{a} \to a^2 \to a^4 \to a^8 \to \mathbf{a}$;

β') $\mathbf{a^3} \to a^6 \to a^{12} \to a^9 \to \mathbf{a^3}$;

γ') $\mathbf{a^5} \to a^{10} \to \mathbf{a^5}$;

δ') $\mathbf{a^7} \to a^{14} \to a^{13} \to a^{11} \to \mathbf{a^7}$;

ϵ') $\mathbf{a^{15}} = 1$;

ζ') $\mathbf{0}$.

If we define the couples (powers of Q, powers of a), the first set of formulas corresponds to (α', δ'), (β', α'), (γ', γ'), (δ', β').
The second set of formulas gives (α', β'), (β', α'), (γ', ϵ'), (δ', δ').

8.5 The case of $\mathbb{F}_8$ (2): optimal Frobenius configurations.

The cyclic configurations found in section 8.3, i.e. the ones leading to polynomials constituted by 18 terms present a very simple structure and lots of connections with the cycle structure of the base field. Unfortunately, the locator polynomials associated to them are *not sparse enough*.
Indeed, our aim is to prove linearity on the growth of the polynomial and the patterns observed in sections 8.3, 8.4 seem to be quadratic: starting from q^2 points we reduce to $\sim \frac{q^2}{2}$ and the terms in the locator polynomials are $\sim \frac{q^2}{4}$, for $q = 8, 16$.

Moreover, from a sparsity point of view, the polynomials we get can be defined "intermediate", but they are not optimal.
Indeed, in the case of $\mathbb{F}_8$, we had the chance to hit a configuration leading to a polynomial made up of 9 terms, by which we deduced a configuration leading to a polynomial made up of 8 terms.
If $\mathbf{Z}_a$ is the set of points leading to the polynomial made up of 9 terms, observing that $F_a + F_b = x_1$, with the above notation, we can deduce that, if $x_1 \in \text{Supp}(F_a)$, then $\mathbf{Z}_b$ leads to a polynomial made up of 8 terms, i.e. *as many terms as the cardinality of the base field*. The configuration[8]

Number of points	Third coordinate
7	0
6	a
3	a^2
3	a^3
3	a^5
2	a^4
2	a^6
2	1

whose configuration list is

$$[1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,1,1,2,2,1,2,2,1,1]$$

gives the locator polynomial $z_1+x_1^6x_2^3+a^3x_1^6x_2^2+a^5x_1^4x_2^2+a^6x_1^6x_2+a^3x_1^2x_2+a^5x_1^3+a^6x_1^2+x_1$, made up of 9 terms (see appendix B, B.3.1 for more details on the configuration).

The locator polynomial contains x_1, so we get the 8 terms configuration, from the set described above. Such a set, corresponds to the configuration list

$$[2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,2,2,2,2,2,1,1,2,1,1,2,2],$$

whose data are described in appendix B.3.4

[8] Notice that, for the cyclical configurations described in the previous section, the choices for the points were univocal. We had to choose 7 points with some third coordinate a, but among the points under consideration, there were only 7 such points. Then, we chose 6 points with some third coordinate b, but among the points under consideration, there were only 6 such points and so on. Thus, all the choices were univocal.
In the case of 9 terms, the choice for the points is not univocal as it was before. Moreover, for some of these choices, we got denser polynomials and so was also for the cyclical permutations of the 9 terms configuration (see for example appendix B, B.3.1 for a permutation giving a denser polynomial).

Definition 8.5.1. An *optimal configuration* in $\mathbb{F}_{2^m}$ is a configuration leading to a polynomial made up of 2^m terms.

The optimal configuration we deduced by the one made up of 9 terms is analogous to the cyclical configurations of the previous section.
Indeed, via some investigations, we could find out that it is invariant for cyclical permutations in the usual sense.
Moreover it presents an interesting structure, somehow connected to the cycles in $\mathbb{F}_8$.
We implemented then a researching algorithm, looking for optimal configurations with an analogous structure and we found out three of them.
We study now the obtained configurations, arranging them in types A,B,C,D and showing the common features of the configurations belonging to the same type.

Type A:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

Table 8.1: Type A configurations in $\mathbb{F}_8$.

Each column in the table above represents a possible choice and the different choices are cyclically permuted as in the intermediate case.
We will explain afterwards why we hightighted in bold the second column.
We use for example type A configurations in order to explain the structure.
The first column is associated to the following configuration list:
$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2.$
The associated polynomial is
$z_1 + a^3x_1^6x_2^2 + a^6x_1^3x_2^2 + x_1^2x_2^2 + a^6x_1^6x_2 + a^5x_2 + a^3x_1^5 + a^5x_1^3$

and the matrix of coefficients (completely analogous to table 8.1) turns out to be:

$$A^{[1]} = \begin{pmatrix} 0 & a^6 & a^3 & 0 \\ a^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^5 & 0 & a^6 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^5 & 0 & 0 \end{pmatrix}$$

The second column corresponds to the list below:
$2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1$
and to the locator polynomial
$z_1 + x_1^6 x_2^2 + x_1^3 x_2^2 + x_1^2 x_2^2 + x_1^6 x_2 + x_2 + x_1^5 + x_1^3$
The coefficient table is

$$A^{[2]} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The general coefficient matrix for type A configuration is

$$A^{[\text{gen}]} = \begin{pmatrix} 0 & A & B & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C & 0 & A & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 \end{pmatrix}$$

and we can find formulas for A, B, C. We take as M the value of the third coordinate appearing once, getting

$$\begin{aligned} A &= M^6 \\ B &= M^3 \\ C &= M^5 \end{aligned} \tag{8.5}$$

so, in analogy with the intermediate configurations, we have the couple of cycles (β, γ).
All the data for Type A configurations are contained in B.3.2.

Let us now focus on the *boldface column*.
As one can easily see by the configuration list, the couples (z_1, z_2) we have chosen are

$(a, 0)$	$(a^2, 0)$	$(a^4, 0)$
(a, a^2)	(a^2, a^4)	(a^4, a)
(a, a^5)	(a^2, a^3)	(a^4, a^6)
$(a, 1)$	$(a^2, 1)$	$(a^4, 1)$
$(a^3, 0)$	$(a^6, 0)$	$(a^5, 0)$
(a^3, a)	(a^6, a^2)	(a^5, a^4)
(a^3, a^4)	(a^6, a)	(a^5, a^2)
(a^3, a^6)	(a^6, a^5)	(a^5, a^3)
$(a^3, 1)$	$(a^6, 1)$	$(a^5, 1)$
$(1, 0)$		

Table 8.2: An optimal Frobenius configuration.

We describe now the properties of the configuration summarized in the above table.
By table 8.1, we impose the third coordinate of all our points to be nonzero. Since for our problem, taken $\mathfrak{b} \in \mathbb{F}_8^*$ we have to choose between two couples of the form (z_1, z_2)[9], namely $(\mathfrak{b}, 0)$ and $(0, \mathfrak{b})$ and 0 cannot be picked as third coordinate, we have to choose $(\mathfrak{b}, 0)$ for each $\mathfrak{b} \in \mathbb{F}_8^*$.
On the other hand, we notice that table 8.1 imposes the third coordinate of one and only one point to be equal to 1. Since, by the above comment, we picked the couple $(1, 0)$, each other couple containing the value 1 has to be of the form $(\mathfrak{b}, 1)$ with $\mathfrak{b} \in \mathbb{F}_8 \setminus \{1, 0\}$.
From the table above, we notice that for the boldface configuration

> if $\mathfrak{b} \in \mathbb{F}_8^*$ is the preferential representative for a cycle in $\mathbb{F}_8$ and we pick the couple $(\mathfrak{b}, c)$, $c \in \mathbb{F}_8^*$, also the couples $(\mathfrak{b}^2, c^2)$, $(\mathfrak{b}^4, c^4)$ have been chosen.

Look at the cycle α, i.e. $\mathbf{a} \to a^2 \to a^4 \to \mathbf{a}$.
The preferential representative of α is a. The choices for $(a^2, *), (a^4, *)$ (i.e. for the four

[9] We recall that z_1, z_2 are respectively the third and the fourth coordinates for our points.

occurrences of a^2, a^4 as third coordinate, required by table 8.1) depend on the four choices made for the couples $(a, \mathfrak{b})$, $\mathfrak{b} \in \mathbb{F}_8$, in the sense we will explain below.
Look for example to the second row of the table, i.e.

$$(a, a^2), (a^2, a^4), (a^4, a).$$

We have
$(a^2, a^4) = ((a)^2, (a^2)^2)$
and
$(a^4, a) = ((a^2)^2, (a^4)^2) = (((a)^2)^2, ((a^2)^2)^2)$.
The same holds also for cycle β, i.e. $\mathbf{a^3} \to a^6 \to a^5 \to \mathbf{a^3}$, whose preferential element is a^3.
We have only made some choices for a^3: the occurrences of a^6, a^5 come "by squarings". For example the row

$$(a^3, a), (a^6, a^2), (a^5, a^4)$$

can be viewed as made of *one independent choice and a couple of squarings*, as shown below:
$(a^6, a^2) = ((a^3)^2, (a)^2)$
$(a^5, a^4) = ((a^6)^2, (a^2)^2) = (((a^3)^2)^2, ((a)^2)^2)$.
The above comments hold also for γ and δ in an obvious way, since $1^2 = 1, 0^2 = 0$.
Let us recall the following

Definition 8.5.2. Let $\mathbb{F}_q$ be a finite field of characteristic p, so that $q = p^n$. The *Frobenius automorphism* is defined as

$$\sigma : \mathbb{F}_q \to \mathbb{F}_q$$

$$a \mapsto a^p.$$

The Frobenius automorphism preserves the cycles:

$$\forall \mathfrak{b} \in \mathbb{F}_8, \ \sigma(\mathfrak{b}) = \mathfrak{c},$$

and $\mathfrak{b}, \mathfrak{c}$ belong to the same cycle.
Moreover, since in a field of characteristic p, $(a + b)^p = a^p + b^p$ and in our case $p = 2$, we can deduce that Frobenius homomorphism preserves syndromes and then it preserves the points' structure.
The Frobenius automorphism is the generator of the cyclic group of the automorphisms in $\mathbb{F}_8$. All these authomorphisms, namely id, σ, σ^2 preserve both the cycles and the syndromes. In our case, i.e. $q = 8$ and $p = 2$, the squaring is simply the application of Frobenius mapping.
Another property of our configuration is

> If $\mathfrak{b} \in \mathbb{F}_8^*$ is the preferential representative for a cycle in $\mathbb{F}_8$, only one couple $(\mathfrak{b}, c)$ with c in the same cycle of $\mathfrak{b}$ has been chosen.

Let us consider for example the quest for couples of the form $(a, \mathfrak{b}),\ \mathfrak{b} \in \mathbb{F}_8$.
In view of the fact that our initial list P contains the couples of points

$$[(a^4, a^4, a, a^2), (a^4, a^4, a^2, a)],$$
$$[(a^2, a^2, a, a^4), (a^2, a^2, a^4, a)],$$
$$[(a, a, a^2, a^4), (a, a, a^4, a^2)],$$

we need to pair off a with some elements of cycle α, so, a priori, we can choose without restriction between

$$(a, a^2), \tag{8.6}$$
$$(a, a^4) \tag{8.7}$$

and we have chosen (a, a^2).
On the other hand, we cannot make both choices in the same configuration, since we would simultaneously have

$$(a, a^2), (a^2, a^4), (a^4, a)$$

and

$$(a, a^4), (a^2, a), (a^4, a^2).$$

This contradicts the requirement to choose *only one* element for each couple in P, which is the first requirement on our configurations, descending from the structure of N and from the Axis of Evil.
For the couples of form $(a, \mathfrak{b}),\ \mathfrak{b} \neq a$, $\mathfrak{b}$ in the cycle α, we have made *one and only one* choice.
By table 8.1, there are four couples of form $(a, \mathfrak{b}),\ \mathfrak{b} \in \mathbb{F}_8$ and we have examined three of them, namely $0, 1$ and a^2. We have to examine the last occurrence of a as third coordinate i.e. the last couple.
Moreover, we know that we cannot choose any other value of $\mathfrak{b}$ neither in α, nor in γ, nor in δ: γ, δ only contain one element and for α we have one and only one available choice.
Then, in order to get the last couple we must pick the last $\mathfrak{b}$ in the cycle β, by elimination. Actually, we have chosen (a, a^5). Another property of **our** optimal configuration is the following

> Consider two distinct cycles θ, θ', such that $\mathfrak{b} \in \mathbb{F}_8^*$ is the preferential element of θ and $c \in \mathbb{F}_8^*$ is the preferential element of θ'. Suppose we have made all the

choices for the couples (b, f), $f \in \mathbb{F}_8$ and to look for the couples (c, d), where $d \in \theta$. The possible values for d depend on the couples of the form (b, e), where e is an element of the cycle θ', as we explain below.

Let us examine then the couples of the form (a^3, b), $b \in \mathbb{F}_8$, in our configuration, which are 5 (see table 8.1). As explained before, we pick $(a^3, 0)$ and $(a^3, 1)$ by table 8.1. Moreover, for (a^3, b), $b \in \mathbb{F}_8$, we have to choose one and only one element of cycle β (different from a^3), exactly as explained for cycle α. The remaining couples, i.e. (a^3, b) with b element of α, turn out to be fixed, depending on the choice made for (a, c), with c element of cycle β.
Indeed, if we choose the couple (a^3, a^2), we have both (a^2, a^3) (coming by the application of the Frobenius mapping to (a, a^5)) and (a^3, a^2) (chosen for a^3) in the same configuration, thing we have excluded.
Then we choose $(a^3, a), (a^3, a^4)$ and we apply Frobenius, getting

$$(a^3, a), (a^6, a^2), (a^5, a^4)$$

$$(a^3, a^4), (a^6, a), (a^5, a^2).$$

Driven by this examination, we give the following

Definition 8.5.3. A *Frobenius configuration* for $\mathbb{F}_8$ is a configuration in $\mathbb{F}_8$ such that

- for each $b \in \mathbb{F}_8^*$, all the couples $(b, 0)$ are in the configuration;
- $(1, 0)$ belongs to the configuration;
- if $b \in \mathbb{F}_8^*$ is the preferential representative for a cycle in $\mathbb{F}_8$ and the couple (b, c), $c \in \mathbb{F}_8^*$, is in the configuration, also the couples (b^2, c^2), (b^4, c^4) do;
- if $b \in \mathbb{F}_8^*$ is the preferential representative for a cycle in $\mathbb{F}_8$, only one couple (b, c) with c in the same cycle of b is in the configuration;
- taken two distinct cycles θ, θ', such that $b \in \mathbb{F}_8^*$ is the preferential element of θ and $c \in \mathbb{F}_8^*$ is the preferential element of θ', suppose we have made all the choices for the couples (b, f), $f \in \mathbb{F}_8$ and to look for the couples (c, d), where $d \in \theta$. The possible values for d are the ones not appearing in the couples (b, e), where e is an element of the cycle θ'.

Definition 8.5.4. A *semi-Frobenius* configuration is a configuration arising from a Frobenius configuration by a cyclical permutation.

For the optimal Frobenius configuration described above, the independent choices we have made are really a few. Most of the couples come by the application of Frobenius mapping and, as explained before, there are some restrictions on the choices.
The couples marked in red in the table above represent these independent choices we have made.
In our investigation, we looked for optimal Frobenius configurations among the Frobenius ones and then for their associated semi-Frobenius configurations. We found three optimal Frobenius configurations.
The boldface columns of type B,C,D are optimal Frobenius configurations while the other ones in the tables are semi-Frobenius configurations. the

Type B:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

Type C:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

Type D:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
4	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
4	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a
5	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
5	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
5	a^5	$\mathbf{a^4}$	a^3	a^2	1	1	a^6

All the data for type B,C,D are in the appendix (see B.3.3,B.3.4,B.3.5).

Remark 8.5.5. As seen above, the formulas for the coefficients of type A configurations are

$$A = M^6$$
$$B = M^3$$
$$C = M^5.$$

The value M is the coordinate appearing only once. Since for the boldface column, i.e. the optimal Frobenius configuration, it holds $M = 1$, we obtain a locator polynomial whose coefficients are all equal to 1
Moreover, every polynomial different from it, has as coefficients 1 and the elements of only one of the other cycles, i.e. α or β.

The configurations type B,C,D behave in the same way of type A configurations. More precisely, there is for each type, one and only one optimal Frobenius configuration satisfying the restrictions above, while the others come by cyclical permutations.

8.6 Optimal Frobenius configurations: what can be generalized?

In this section, we give some partial results on the generalization of the optimal Frobenius and semi-Frobenius configurations to fields larger than $\mathbb{F}_8$.
For this purpose, we first recall that the cycles in $\mathbb{F}_8$ are

$\alpha)$ $\mathbf{a} \to a^2 \to a^4 \to \mathbf{a}$;

$\beta)$ $\mathbf{a^3} \to a^6 \to a^5 \to \mathbf{a^3}$;

$\gamma)$ $\mathbf{a^7} = 1$;

$\delta)$ $\mathbf{0}$.

In this case, the order of the Frobenius mapping is 3.
The cycles in $\mathbb{F}_{16}$ are:

$\alpha')$ $\mathbf{a} \to a^2 \to a^4 \to a^8 \to \mathbf{a}$;

$\beta')$ $\mathbf{a^3} \to a^6 \to a^{12} \to a^9 \to \mathbf{a^3}$;

$\gamma')$ $\mathbf{a^5} \to a^{10} \to \mathbf{a^5}$;

$\delta')$ $\mathbf{a^7} \to a^{14} \to a^{13} \to a^{11} \to \mathbf{a^7}$;

$\epsilon')$ $\mathbf{a^{15}} = 1$;

$\zeta')$ $\mathbf{0}$.

and the order of the Frobenius mapping is 4, so we have $id, \sigma, \sigma^2, \sigma^3$. Notice that Frobenius mapping preserves both cycles and syndromes, so it preserves the structure of the points.
The cycle structure of $\mathbb{F}_{16}$ is rather different than the one of $\mathbb{F}_8$ and it influences our possibilities in constructing points configurations with analogous restrictions as the optimal Frobenius configurations of the previous section.
First of all, we notice that the length of cycles in $\mathbb{F}_{p^m}$ divides m. In the case of $\mathbb{F}_8$, $m = 3$ and the cycles have length 1 (the trivial ones) and 3. In the case of $\mathbb{F}_{16}$, $m = 4$ and the lengths of the cycles are $1, 2, 4$.
Consider first cycle γ'. Clearly, we have to pair off two elements of γ'. More precisely, we must have either (a^5, a^{10}) or (a^{10}, a^5). But if we choose (a^5, a^{10}) and we apply as for $\mathbb{F}_8$ the Frobenius mapping we get the couple (a^{10}, a^5). But, in our problem, we have escluded the occurrence of both these couples. So this "degenerate short cycle" is not compatible with the application of Frobenius mapping as in $\mathbb{F}_8$ and it makes necessary to change the way to generate a configuration.

But there is something more. Consider for example the couples of the form $(a, \mathfrak{b})$. We have to deal with those for which $\mathfrak{b}$ is an element in cycle α'. If we choose for example the couple (a, a^2), by the application of Frobenius mapping, we get

$$(a, a^2), (a^2, a^4), (a^4, a^8), (a^8, a)$$

and, similarly, for (a, a^8)

$$(a, a^8), (a^2, a), (a^4, a^2), (a^8, a^4).$$

Clearly we cannot choose both these couples[10], but we notice that, if we pick only one of them, we do not deal respectively with $(a, a^4), (a^4, a)$ and $(a^2, a^8), (a^8, a^2)$, so we do not treat all the couples of elements in α'. Anyway, if we pick the couple (a, a^4) and we apply as usual the Frobenius mapping we get

$$(a, a^4), (a^2, a^8), (a^4, a), (a^8, a^2),$$

which is incompatible with our usual requirement on the couples: we only want to choose one and only one between $(\mathfrak{a}, \mathfrak{b})$ and $(\mathfrak{b}, \mathfrak{a})$.
This problem clearly occurs also for β' and δ'.
We can relate the problem to the theory of permutations. Consider cycle α' and suppose to make the following choice for $(a, \mathfrak{b})$, with $\mathfrak{b}$ in α', applying Frobenius as usual:

$$(a, a^2), (a^2, a^4), (a^4, a^8), (a^8, a).$$

Such a choice can be seen as a cyclical permutation of α', i.e.

$$\begin{pmatrix} a & a^2 & a^4 & a^8 \\ a^2 & a^4 & a^8 & a \end{pmatrix} = (a, a^2, a^4, a^8) = \lambda.$$

Making another such choice for a (i.e. pairing a with another element of cycle α) and applying Frobenius means taking a power of λ.
Now, the i-th power of a cycle of length m is a cycle $\Leftrightarrow GCD(i, m) = 1$.
If such a permutation is not a cycle, is a product of disjoint cycles of the same length.
If the permutation is a cycle (as λ), then we cannot find in the application of Frobenius both the couples of the form $(\mathfrak{a}, \mathfrak{b})$ and $(\mathfrak{b}, \mathfrak{a})$: if it was so, the permutation would be the product of disjoint transpositions, so it would not be cyclic.
So the cases presenting some problems w.r.t. Frobenius applications are the ones corresponding to powers of permutations which are products of disjoint transpositions.
This cannot happen for $\mathbb{F}_8$, since m is a prime number, whereas it is exactly what happens for $\mathbb{F}_{16}$.
We can overcome the problem of finding both the couples of the form $(\mathfrak{a}, \mathfrak{b})$ and $(\mathfrak{b}, \mathfrak{a})$ while pairing off elements of the same cycle, by admitting two distinct kinds of application of the Frobenius mapping for $\mathbb{F}_{16}$: *short* and *long* applications.
A long application is the analogous of what done in $\mathbb{F}_8$, i.e., given a couple $(\mathfrak{b}, \mathfrak{c})$ we compute $(\sigma(\mathfrak{b}), \sigma(\mathfrak{c}))$, ..., $(\sigma^4(\mathfrak{b}), \sigma^4(\mathfrak{c}))$.
A short application admits only the couples $(\mathfrak{b}, \mathfrak{c})$ and $(\sigma(\mathfrak{b}), \sigma(\mathfrak{c}))$ ($(\sigma^2(\mathfrak{b}), \sigma^2(\mathfrak{c})) = (\mathfrak{c}, \mathfrak{b})$): it

[10] It is similar to what done with $\mathbb{F}_8$: we would contradict the requirement to have only one couple between $(\mathfrak{a}, \mathfrak{b})$ and $(\mathfrak{b}, \mathfrak{a})$.

means that, when the permutation is not a cycle, we consider the couples corresponding to the distinct cycles in which it is decomposed[11].
For α', for example, we can take the long application

$$(a, a^2), (a^2, a^4), (a^4, a^8), (a^8, a),$$

$$\begin{pmatrix} a & a^2 & a^4 & a^8 \\ a^2 & a^4 & a^8 & a \end{pmatrix} = (a, a^2, a^4, a^8) = \lambda.$$

jointed with the short application

$$(a, a^4), (a^2, a^8).$$

$$\begin{pmatrix} a & a^2 & a^4 & a^8 \\ a^4 & a^8 & a & a^4 \end{pmatrix} = (a, a^4), (a^2, a^8) = \lambda^2.$$

This way, we can pair a with all the other elements in α, without getting both the couples of the form (a, b) and (b, a), situation we have excluded.
Clearly, the problem can only arise for couples (b, c), b, c in the same cycle.
Consequently, the kind of search we are developing now (still in progress) is to check the configurations obtained by choosing the couples and applying long and short applications in a consistent way, in order to get the analogous of a Frobenius configuration for $\mathbb{F}_{16}$.
There are many types of such configurations, we will start with the type related to the following table, only because, in analogy with the tables for type A,B,C,D in $\mathbb{F}_8$, it involves only 2 consecutive numbers and 1 (see the "total" column). In this table, we have counted the couples arising from short applications, long applications and no applications of Frobenius (as explained above for γ'). The total m displayed in the table for a certain power a^i means that there are m couples of the form (a^i, b). The number displayed in the "long" cell is the number of such couples arising by a long Frobenius application on an independent choice and the one displayed in the "short" cell is the number of couples arising by a short Frobenius application[12].
The line of $a^{15} = 1$ and the one relative to an element in γ' are particular since there can be "no Frobenius applications".
We remark that the short applications of the Frobenious mapping are related only to couples (a^i, a^j) such that a^i, a^j belong to the same cycle.
Finally, consider the cycle structure in $\mathbb{F}_{32}$:

$\alpha'')$ $\mathbf{a} \to a^2 \to a^4 \to a^8 \to a^{16} \to \mathbf{a}$;

[11] For finite fields $\mathbb{F}_{2^m}$, m not a prime number, we need to study the cyclic structure of permutations, in order to find the corresponding configurations.
[12] Clearly they represent the choice we make if we are looking at the row of a preferential representative!

$a^?$	Long	Short	No	Total
a	8	1	0	9
a^2	8	1	0	9
a^3	8	1	0	9
a^4	8	0	0	8
a^5	8	0	1	9
a^6	8	1	0	9
a^7	8	1	0	9
a^8	8	0	0	8
a^9	8	0	0	8
a^{10}	8	0	0	8
a^{11}	8	0	0	8
a^{12}	8	0	0	8
a^{13}	8	0	0	8
a^{14}	8	1	0	9
$a^{15} = 1$	0	0	1	1

Table 8.3: Generalization to $\mathbb{F}_{16}$.

β'') $\mathbf{a^3} \to a^6 \to a^{12} \to a^{24} \to a^{17} \to \mathbf{a^3}$;

γ'') $\mathbf{a^5} \to a^{10} \to a^{20} \to a^9 \to a^{18} \to \mathbf{a^5}$;

δ'') $\mathbf{a^7} \to a^{14} \to a^{28} \to a^{25} \to a^{19} \to \mathbf{a^7}$;

ϵ'') $\mathbf{a^{11}} \to a^{22} \to a^{13} \to a^{26} \to a^{21} \to \mathbf{a^{11}}$;

ζ'') $\mathbf{a^{15}} \to a^{30} \to a^{29} \to a^{27} \to a^{23} \to \mathbf{a^{15}}$;

η'') $\mathbf{a^{31}} = 1$;

θ'') $\mathbf{0}$.

Here, the cycles have the same structure as $\mathbb{F}_8$. All the cycles (excluded the cycles of 0 and 1) have the same length. Moreover, in this case $m = 5$ is a prime number, so all the powers of a cycle are cycles: we do not need short and long applications.
Now, we are verifying the behaviour of generalized configurations to $\mathbb{F}_{16}$ and $\mathbb{F}_{32}$.

More precisely, again in analogy with $\mathbb{F}_8$, we are dealing with this table for $\mathbb{F}_{32}$:

$a^?$	Number of points
a	19
a^2	19
a^3	18
a^4	19
a^5	17
a^6	18
a^7	16
a^8	19
a^9	17
a^{10}	17
a^{11}	15
a^{12}	18
a^{13}	15
a^{14}	16
a^{15}	14
a^{16}	19
a^{17}	18
a^{18}	17
a^{19}	16
a^{20}	17
a^{21}	15
a^{22}	15
a^{23}	14
a^{24}	18
a^{25}	16
a^{26}	15
a^{27}	14
a^{28}	16
a^{29}	14
a^{30}	14
$a^{31} = 1$	1

We believe we will be able to find optimal Frobenius configurations among the configurations described in the tables above.

Bibliography

[1] M.E. Alonso, M.G. Marinari, T. Mora, *The big Mother of all Dualities : Möller algorithm.*, Comm. Alg. **31**(2) (2003), 783–818; 374–383.

[2] M.E. Alonso, M.G. Marinari, T. Mora, *The big Mother of all Dualities 2: Macaulay Bases*, Applicable Algebra in Engineering, Communication and Computing archive Vol. **17**, Issue 6, November 2006, 409–451.

[3] D. Augot, M. Bardet, and J.-C. Faugère, *Efficient decoding of (binary) cyclic codes above the correction capacity of the code using Groebner bases*, Proc. of ISIT 2003, 2003, 362.

[4] D. Augot, M. Bardet, and J.-C. Faugère, *On formulas for decoding binary cyclic codes*, Proc. of ISIT 2007, 2007, 2646–2650.

[5] A.M. Barg, E. Krouk, and H. C. A. van Tilborg, *On the complexity of minimum distance decoding of long linear codes*, IEEE Trans. on Inf. Th. **45** (1999), no. 5, 1392–1405

[6] E.R. Berlekamp, *Binary BCH codes for correcting multiple errors*, Algebraic Coding Theory, McGraw-Hill, New York 1968

[7] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. on Inf. Th. **24** (1978), no. 3, 384–386

[8] C. Bertone, F. Cioffi, P. Lella, M. Roggero, *Upgraded methods for the effective computation of marked schemes on a strongly stable ideal*, J. Symb. Comput. (2012), http://dx.doi.org/10.1016/j.jsc.2012.07.006

[9] C. Bertone, F. Cioffi, M. Roggero, *A division algorithm in an affine framework for flat families covering Hilbert schemes* , arXiv:1211.7264 [math.AC]

[10] C. Bertone, P. Lella, M.Roggero, *A Borel open cover of the Hilbert scheme*, J. Symbolic Comput. **53** (2013), 119–135.

[11] B. Buchberger, *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, in Bose N.K. (Ed.) *Multidimensional Systems Theory* (1985), 184–232, Reider

[12] B. Buchberger, H.M. Moeller, *The construction of multivariate polynomials with preassigned zeros*, Lec. Not. in Computer Science, Volume **144**, 1982, pp 24–31.

[13] M. Caboara, T. Mora *The Chen-Reed-Helleseth-Truong decoding algorithm and the Gianni-Kalkbrenner Groebner shape theorem*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), no. 3, 209–232

[14] E. Cartan *Sur l'intégration des systèmes d'équations aux différentielles totals.* Ann. Éc. Norm. 3^e série **18** (1901) 241.

[15] E. Cartan *Sur la structure des groupes infinis de transformations.* Ann. Éc. Norm. 3^e série **21** (1904) 153.

[16] E. Cartan *Sur les systèmes en involution d'équations aux dérivées partielles du second ordre à une fonction inconnue de trois variables indépendentes.* Bull. Soc. Marth. **39** (1920) 356.

[17] M. Ceria, `JMBTest.lib`. A library for Singular which performs JM basis test. (2012).

[18] M. Ceria, `JMSConst.lib`. A library for Singular which constructs J-Marked Schemes. (2012).

[19] M.Ceria, T.Mora and M.Roggero, *Term-ordering free involutive bases*, arXiv:1310.0916.

[20] L. Cerlienco, M. Mureddu, *Algoritmi combinatori per l'interpolazione polinomiale in dimensione* ≥ 2, preprint(1990).

[21] L. Cerlienco, M. Mureddu, *From algebraic sets to monomial linear bases by means of combinatorial algorithms*, Discrete Math. **139**, 73–87.

[22] L. Cerlienco, M. Mureddu, *Multivariate Interpolation and Standard Bases for Macaulay Modules*, J. Algebra **251** (2002), 686–726.

[23] X. Chen, I. S. Reed, T. Helleseth and T. K. Truong, *General principles for the algebraic decoding of cyclic codes*, EEE Trans. on Inf. Th. **40** (1994a), 1661–1663.

[24] X. Chen, I. S. Reed, T. Helleseth and T. K. Truong, *Use of Groebner bases to decode binary cyclic codes up to the true minimum distance*, IEEE Trans. on Inf. Th. **40** (1994c), no. 5, 1654–1661

[25] X. Chen, I. S. Reed, T. Helleseth and T. K. Truong, *Algebraic decoding of cyclic codes: a polynomial ideal point of view*, Contemp. Math., Vol. **168**, Amer. Math. Soc., Providence, 1994b, 15–22.

[26] F.Cioffi, P. Lella, M.G. Marinari, M. Roggero, *Segments and Hilbert schemes of points*, Disc. Math. **311**, 20 (2011), 2238–2252.

[27] F. Cioffi, M. Roggero, *Flat families by strongly stable ideals and a generalization of Gröbner bases*, J. Symb. Comput, Vol. **46**, Issue 9, September 2011, 1070–1084.

[28] A.B. III Cooper, *Direct solution of BCH decoding equations*, Comm., Cont. and Sign. Proc. (1990), 281–286.

[29] A.B. III Cooper, *Finding BCH error locator polynomials in one step*, Electronic Letters **27** (1991), no. 22, 2090–2091.

[30] W. Decker, G.-M. Greuel, G. Pfister, H. Schönemann: SINGULAR 3-1-4 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de (2012).

[31] T.Deery, *Rev-lex segment ideals and minimal Betti numbers,* The Curves Seminar at Queen's, Vol. X, (Kingston, ON, 1995), Queen's Papers in Pure and Appl. Math., Vol. **102**, Queen's Univ., Kingston, ON, 1996, 193–219.

[32] Delassus E., *Extension du théorème de Cauchy aux systèmes les plus généraux d'équations aux dérivées partielles*. Ann. Éc. Norm. 3^e série **13** (1896) 421–467

[33] S. Eliahou and M. Kervaire. *Minimal resolutions of some monomial ideals,* J. Algebra, 129(1):1-25, 1990.

[34] J. Erdös, *On the structure of ordered real vector spaces*, Publ. Math. Debrecen **4** (1956), 334–343

[35] J. Farr, S. Gao, *Computing Groebner Bases for Vanishing Ideals of Finite Sets of Points*, 16th International Symposium, AAECC-16, (2004).

[36] J.C. Faugere, P. Gianni, D. Lazard, T. Mora, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, J. Symb. Comp., **16** (1993) 329–344.

[37] B. Felszeghy, B. Ráth, L. Rónyai *The Lex Game and some applications*, J. Symbolic Computation **41** (2006), 663–681.

[38] A. Galligo, *Théoreme de division et stabilité en géométrie analytique locale,* Ann. Inst. Fourier (Grenoble) **29** (1979), no. 2, vii, 107–184.

[39] S. Gao, V.m. Rodrigues, J. Stroomer, *Groebner basis structure of finite sets of points*, preprint.

[40] Gerdt V.P. *Involutive Algorithms for Computing Groebner Bases*. In *Computational Commutative and Non-Commutative Algebraic Geometry*, S.Cojocaru, G.Pfister and V.Ufnarovski (Eds.), NATO Science Series, IOS Press, (2005), 199–225

[41] Gerdt V.P., Blinkov Y.A. *Involutive bases of Polynomial Ideals*, Math. Comp. Simul. **45** (1998), 543–560

[42] Gerdt V.P., Blinkov Y.A. *Minimal involutive bases*, Math. Comp. Simul. **45** (1998), 519–541

[43] M. L. Green, *Generic initial ideals*, Six lectures lectures on commutative algebra (Bellaterra, 1996), Progr. Math., Vol. 166, Birkhäuser, Basel, 1998, 119–186.

[44] Gianni P., *Properties of Gröbner Bases under Specialization*, L. N. Comp. Sci. **378** (1987), 293–297, Springer.

[45] R.L. Graham, M. Groetschel, L. Lovàsz, *Handbook of Combinatorics.*, Vol. 1, The MIT Press, Cambridge, Massachusetts, 1995.

[46] Gunther, N. *Sur la forme canonique des systèmes déquations homogènes* (in russian) [Journal de l'Institut des Ponts et Chaussées de Russie] Izdanie Inst. Inž. Putej Soobščenija Imp. Al. I. **84** (1913) .

[47] Gunther, N. *Sur la forme canonique des equations algébriques* C.R. Acad. Sci. Paris **157** (1913), 577–80

[48] A. Hashemi , M. Schweinfurter, W. M. Seiler, *Quasi-Stability versus Genericity*, Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, Volume 7442, 2012, 172-184.

[49] G. Hegedus, L. Rónyai, *Standard monomials for q-uniform families and a conjecture of Babai and Frankl*. Central European Journal of Mathematics 1, 198-207.

[50] Hilbert D., *Uber die Theorie der algebraicschen Formen*, Math. Ann. **36** (1890), 473–534

[51] Hironaka, H. *Idealistic exponents of singularity* In: *Algebraic Geometry, The Johns Hopkins Centennial Lectures* (1977) 52-125

[52] D. G. Hoffman, *Coding theory: The essential*, Dekker, New York, 1991

[53] W. C. Huffman, V. Pless, *Fundamentals of Error-correcting Codes*, Cambridge University Press, 2003.

[54] M. Janet, *Sur les systèmes d'équations aux dérivées partelles*, J. Math. Pure et Appl., 3, (1920), 65-151.

[55] M. Janet, *Les modules de formes algébriques et la théorie générale des systemes différentiels*, Annales scientifiques de l'École Normale Supérieure, 1924.

[56] M. Janet, *Les systèmes d'équations aux dérivées partelles*, Gauthier-Villars, 1927.

[57] M. Janet, *Lecons sur les systèmes d'équations aux dérivées partelles* , Gauthier-Villars.

[58] A.E. Kézdy, H.S. Snevily, *Polynomials that vanish on distinct nth roots of unity.* Combinatorics, Probability and Computing **13**, 37-59.

[59] M. Kalkbrenner, *Solving Systems of Algebraic Equations by Using Groebner Bases*, L. N. Comp. Sci. 378 (1987), pagg. 282-292, Springer.

[60] M. Kalkbrener, *On the stability of Gröbner Bases under specialization*, J. Symb. Comp. **24** (1997), 51–58

[61] D.E. Knuth, *The art of computer programming*, vol 3., Addison-Wesley Publishing Company.

[62] D. Lazard, *Ideal Basis and Primary Decomposition: Case of two variables*, J. Symb. Comp. 1 (1985), 261-270.

[63] M. Lederer, *The vanishing ideal of a finite set of closed points in affine space*, Journal of Pure and Appliead Algebra 212, (2008), pagg. 1116-1133.

[64] P. Lella, M. Roggero, *Rational components of Hilbert schemes*, Rendiconti del Seminario Matematico dell'Università di Padova.

[65] R. Lidl, H. Niederreiter, *Finite Fields*, Volume 20, Parte 1 Volume 20 di Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1997

[66] P. Loustaunau, E. V. York, *On the decoding of cyclic codes using GrÃűbner bases*, AAECC 8 (1997), no. 6, 469–483

[67] S. Lundqvist, *Vector space bases associated to vanishing ideals of points*, Journal of Pure and Applied Algebra, 214, Issue 4, (2010), pagg 309-321.

[68] F.S. Macaulay, *Some properties of enumeration in the theory of modular systems,* Proc. London Math Soc, 26, (1927)

[69] M.G. Marinari and Teo Mora, *Cerlienco-Mureddu Correspondence and Lazard Structural Theorem.*, Revista Investicación Operacional, Vol.27, No.2, 155-178, 2006.

[70] M.G. Marinari and Teo Mora, *A remark on a remark by Macaulay or Enhancing Lazard Structural Theorem.*, Bulletin of the Iranian Mathematical Society Vol. 29 No. 1 (2003), pagg. 1-45.

[71] M.G. Marinari and Teo Mora, *Some Comments on Cerlienco-Mureddu Algorithm and Enhanced Lazard Structural Theorem*, Rejected by ISSAC-2004 (2004).

[72] M.G. Marinari, H.M Moeller, T. Mora, *Groebner Duality*, Quaderno del Dipartimento di Matematica dell'Università di Genova.

[73] M.G. Marinari, H.M Moeller, T. Mora, *Groebner Bases of Ideals Defined by Functionals with an Application tp Ideals of Projective Points*, Applicable Algebra in Engineering, Communication and Computing, vol. 4, 1993, Springer.

[74] M.G. Marinari, T. Mora, *The Axis-of-Evil theorem.*, 2008, avaliable on the website http://www.disi.unige.it/person/MoraF/publications.html

[75] M.G . Marinari, L. Ramella *Borel Ideals in three variables*, Beiträge zur Algebra und Geometrie. Contributions to Algebra and Geometry, Vol 47 (2006), N. 1, 437-446.

[76] F. Mora, *De Nugis Groebnerialium 2: Applying MacaulayâĂŹs Trick in Order to Easily Write a Groebner Basis*, Applicable Algebra in Engineering, Communication and Computing archive Vol. 13 , 2003, 409-451.

[77] T, Mora, L. Perret, S. Sakata, M. Sala, C. Traverso, *Groebner Bases, Coding, and Cryptography*, Springer, 2009.

[78] T. Mora, E. Orsini, M. Sala, *General error locator polynomials for binary cyclic codes with* $t \leq 2$ *and* $n < 63$, BCRI preprint, `www.bcri.ucc.ie` 43, UCC, Cork, Ireland, 2006.

[79] T. Mora, *Solving polynomial equation systems: Macaulay's paradigm and Groebner technology*, Cambridge University Press, 2005.

[80] A. Muratović-Ribić, Q.Wang *Partitions and compositions over finite fields*, arXiv preprint arXiv:1205.4250, 2012.

[81] G.H. Norton, A. Sălăgean, *Strong Gröbner bases for polynomials over a principal ideal ring*, Bull. Austral. Math. Soc. **64** (2001), 505-528.

[82] E. Orsini and M. Sala, *Correcting errors and erasures via the syndrome variety*, J. Pure Appl. Algebra 200 (2005), 191–226.

[83] E. Orsini and M. Sala, *General error locator polynomials for binary cyclic codes with* $t \leq 2$ *and* $n < 63$, IEEE Trans. on Inf. Th. 53 (2007), 1095–1107.

[84] J. F. Pommaret, *Systems of partial differential equations and Lie pseudogroups*, Gordon and Brach (1978)

[85] J. F. Pommaret, Akli H. *Effective Methods for Systems of Algebraic Partial Differential Equations*, Progress in Mathematics **94** (1990), 411–426, Birkhäuser

[86] A. Reeves and B. Sturmfels, *A note on polynomial reduction*, J. Symbolic Comput. 16 (1993), n.3, 273-277.

[87] C. Riquier, *De l'existence des intégrales dans un système differentiel quelconque* Ann. Éc. Norm. 3^e série **10** (1893) 65–86.

[88] C. Riquier, *Sur une questione fondamentale du Calcul intégral* Acta mathematica **23** (1899), 203

[89] C. Riquier, *Les systèmes d'équations aux dérivées partielles* (1910), Gauthiers-Villars.

[90] L. Robbiano *Term orderings on the polynomial ring* L. N. Comp. Sci. **204** (1985), 513–7, Springer

[91] L.B. Robinson, *Sur les systémes d'équations aux dérivées partialles* C.R. Acad. Sci. Paris **157** (1913), 106–108

[92] L.B. Robinson, *A new canonical form for systems of partial differential equations* American Journal of Math. **39** (1917), 95–112

[93] M. Sala, *Groebner basis techniques to compute weight distributions of shortened cyclic codes*, preprint.

[94] F.O. Schreyer, *A standard basis approach to syzygies of canonical curves*, J. Reine angew. Math. **421** (1991), 83–123

[95] F. Schwartz, *Reduction and Completion Algorithm for Partial Differential Equations*, Proc. ISSA'92 (1992), 49–56 ACM

[96] C.E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27** (1948), 379–423, 623–656.

[97] W. M. Seiler, *Involution - The Formal Theory of Differential Equations and its Applications in Computer Algebra and Numerical Analysis* Habilitation Thesis (2002)Universität Mannheim, http://www.mathematik.uni-kassel.de/ seiler/Papers/habil.html

[98] W. M. Seiler, *Involution - The Formal Theory of Differential Equations and its Applications in Computer Algebra*, Springer-Verlag, Berlin/Heidelberg 2010, Algorithms and Computation in Mathematics, Vol. 24.

[99] W. M. Seiler, *A Combinatorial Approach to Involution and Delta-Regularity I: Involutive Bases in Polynomial Algebras of Solvable Type* , Applicable Algebra in Engineering, Communication and Computing, 20 (2009), 207-259

[100] W. M. Seiler, *A Combinatorial Approach to Involution and Delta-Regularity II: Structure Analysis of Polynomial Modules with Pommaret Bases*, Applicable Algebra in Engineering, Communication and Computing, 20 (2009) 261-338.

[101] R.P. Stanley, *Enumerative Combinatorics,* Volume I, Cambridge University Press, 2011.

[102] R.P. Stanley, *Enumerative Combinatorics,* Volume II, Cambridge University Press, 1999.

[103] S. Steidel, `pointid.lib`. Procedures for computing a factorized lex GB of the vanishing ideal of a set of points via the Axis-of-Evil Theorem (M.G. Marinari, T. Mora) (2011).

[104] Trinks W., *Über B. Buchberger Verfahren, Systeme algebraischer Gleichungen zu lösen*, J. Numb. Th. **10** (1978), 475-488

[105] J. H. Van Lint, *Coding Theory*, Springer-Verlag, Berlin, Heidelberg, New York 1973

[106] A. Vardy, *Algorithmic complexity in coding theory and the minimum distance problem*, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, 1997, 92–109

[107] A. Zarkov, Y. Blinkov, *Involution Approach to Investing Polynomial Systems*, Math. Comp. Simul. **42** (1996), 323–332

[108] A. Zarkov, *Solving zero-dimensional involutive systems*, Progress in Mathematics **143** (1996), 389–399, Birkhäuser

APPENDIX A

Singular code of the libraries.

We display in this section the source code of the Singular libraries presented in section 6.2 of chapter 6.
The libraries available with Singular 3-1-6:
`http://www.singular.uni-kl.de/index.php/singular-download.html`
Their documentation can be found at:
`http://www.singular.uni-kl.de/Manual/latest/sing_2017.htm#SEC2093` and
`http://www.singular.uni-kl.de/Manual/latest/sing_2022.htm#SEC2098`
All the source code presented here is also available online at
`http://www.singular.uni-kl.de/svn/trunk/Singular/LIB/JMBTest.lib` and
`http://www.singular.uni-kl.de/svn/trunk/Singular/LIB/JMSConst.lib`

A.1 JMBTest.lib: a test for J-marked bases.

```
1 ////////////////////////////////////////////////////////////////////////////
2 /* version="$Id:$";
3 category="Algebraic Geometry";
```

```
// summary description of the library
info="
LIBRARY: JMBTest.lib A library for Singular which performs JM basis test.
AUTHOR: Michela Ceria, email: michela.ceria@unito.it

SEE ALSO: JMSConst_lib
KEYWORDS: J−marked schemes

OVERVIEW:
The library performs the J−marked basis test, as described in [CR], [BCLR].
Such a test is performed via the criterion explained in [BCLR],
concerning Eliahou−Kervaire polynomials (EK from now on).
We point out that all the polynomials are homogeneous
and they must be arranged by degree.
The fundamental steps are the following:@∗
−construct the Vm polynomials, via the algorithm VConstructor
explained in [CR];@∗
−construct the Eliahou−Kervaire polynomials defined in [BCLR];@∗
−reduce the Eliahou−Kervaire polynomials using the Vm's;@∗
−if it exist an Eliahou−Kervaire polynomial such that its reduction
mod Vm is different from zero, the given one is not a J−Marked basis.

The algorithm terminates only if the ordering is rp.
Anyway, the number of reduction steps is bounded.

REFERENCES:
[CR] Francesca Cioffi, Margherita Roggero,Flat Families by Strongly
Stable Ideals and a Generalization of Groebner Bases,
J. Symbolic Comput. 46, 1070−1084, (2011).@∗
[BCLR] Cristina Bertone, Francesca Cioffi, Paolo Lella,
Margherita Roggero, Upgraded methods for the effective
computation of marked schemes on a strongly stable ideal,
Journal of Symbolic Computation
(2012), http://dx.doi.org/10.1016/j.jsc.2012.07.006 @∗ ∗/
/∗PROCEDURES:
  Minimus(ideal) minimal variable in an ideal
  Maximus(ideal) maximal variable in an ideal
  StartOrderingV(list,list) ordering of polynomials as in [BCLR]
  TestJMark(list) tests whether we have a J−marked basis
```

```
"; */
LIB "qhmoduli.lib";
LIB "monomialideal.lib";
LIB "ring.lib";
////////////////////////////////////////////////////////////////////////
 proc mod_init()
/*"USAGE: mod_init();
RETURN: struct: jmp
EXAMPLE: example mod_init; shows an example"*/
{
newstruct("jmp", "poly␣h,␣poly␣t");
}
example
{ "EXAMPLE:"; echo = 2;
  mod_init();
}
////////////////////////////////////////////////////////////////////////
 proc Terns(list G, int c)
/*"USAGE: Terns(G,c); G list, c int
RETURN: list: T
NOTE: Input is a list of J−marked polynomials
(arranged by degree) and an integer.
EXAMPLE: example Terns; shows an example"*/
{
list T=list();
int z;
 for(int k=1; k<=size(G[c]);k=k+1)
    {
//Loop on G[c] making positions of polynomials in G[c]
       z=size(T);
       T=insert(T,list(1,c,k) ,size(T));
    }
return(T);
}
//example
{ "EXAMPLE:"; echo = 2;
ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
```

```
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
Terns(G2F, 1);
Terns(G2F, 2);
}
////////////////////////////////////////////////////////////////////////
 proc VConst(list G, int c)
/*"USAGE: VConst(G, c); G list, c int
RETURN: list: V
NOTES: this procedure computes the Vm polynomials following the
algorithm in [CR],but it only keeps in memory the monomials by
which the G's must be multplied and their positions.
EXAMPLE: example VConst; shows an example"*/
{
jmp f=G[1][1];
int aJ=deg(f.h);
// minimal degree of polynomials in G
//print(aJ);
list V=list();
V[1]=Terns(G,1);
// V[1]=G[1] (keeping in memory only [head, position])
//print(c-aJ+1);
int i;
int j;
int m;
list OO;
jmp p;
  for(m=2; m<=c-aJ+1; m=m+1)
  {
      //print("entro nel form");
```

```
121       if(m>size(G))
122           {V[m]=list();
123 //If we have not G[m] we insert a list()
124           //print("vuota prima");
125           }
126       else
127           {V[m]=Terns(G,m);
128           //print("piena prima");
129           }
130       for(i=1; i<nvars(basering)+1; i=i+1)
131           {
132             //print("entrata fori");
133             //print(i);
134             for(j=1; j<=size(V[m-1]); j=j+1)
135                 {
136                         p=G[V[m-1][j][2]][V[m-1][j][3]];
137                       //print(p.h);
138                       //print(p.t);
139                       //print(var(i));
140                       //print(Minimus(V[m-1][j][1]*p.h));
141                   if(var(i)<=Minimus(variables(V[m-1][j][1]*p.h)))
142                       {
143 //Can I multiply by the current variable?
144                         //print("minoremin");
145                         //print("fin qui ci sono");
146                          //print(V[m-1][j][1]);
147                         OO=list(var(i)*V[m-1][j][1],V[m-1][j][2],V[m-1][j][3]);
148                       V[m]=insert(V[m], OO ,size(V[m]));
149                       }
150                 }
151           }
152   }
153 return (V);}
154 //example
155 { "EXAMPLE:"; echo = 2;
156   ring r=0, (x,y,z), rp;
157 jmp r1;
158 r1.h=z^3;
159 r1.t=poly(0);
```

```
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
VConst(G2F,4,basering);}
////////////////////////////////////////////////////////////////////
proc Minimus(ideal L)
/*"USAGE: Minimus(L); G list, c int
RETURN: list: V
NOTES: it returns the minimal variable generating the ideal L.@*
The input must be an ideal generated by variables.
EXAMPLE: example Minimus; shows an example"*/
{
poly min=L[1];
int i;
for(i=2;i<=size(L); i++)
   {
      if(L[i]<min){min=L[i];}
   }
return(min);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z), rp;
ideal I=y,x,z;
Minimus(I);
}
////////////////////////////////////////////////////////////////////
proc Maximus(ideal L)
/*"USAGE: Maximus(L); G list, c int
RETURN: list: V
NOTES: it returns the maximal variable generating the ideal L.@*
The input must be an ideal generated by variables.
```

```
199 EXAMPLE: example Maximus; shows an example"*/
200 {
201 poly max=L[1];
202 int i;
203 for(i=2;i<=size(L); i++)
204   {
205     if(L[i]>max){max=L[i];}
206   }
207 return(max);
208 }
209 //example
210 { "EXAMPLE:"; echo = 2;
211  ring r=0, (x,y,z), rp;
212 ideal I=y,x,z;
213 Maximus(I);
214 }
215 ////////////////////////////////////////////////////////////////////////
216  proc GJmpMins(jmp P, jmp Q)
217 /*"USAGE: GJmpMins(P,Q); P jmp, Q jmp
218 RETURN: int: d
219 EXAMPLE: example GJmpMins; shows an example"*/
220 {
221   int d=1;
222 //−1=lower, 0=equal, 1=higher
223 //At the beginning suppose Q is higher
224   if(deg(P.h)<deg(Q.h))
225       {
226 //Compare degrees;
227        d=−1;
228        //print("Per Grado");
229       }
230   if(deg(P.h)==deg(Q.h))
231       {
232        if(P.h==Q.h)
233            {
234             if(P.t==Q.t)
235                  {
236 //head=tail
237                   d=0;
```

```
238                     //print("Uguali");
239                    }
240              }
241         else
242              {
243 //print(Minimus(variables(P.h/gcdMon(P.h,Q.h))));
244 //print(Minimus(variables(Q.h/gcdMon(P.h,Q.h))));
245
246 if(Minimus(variables(P.h/gcdMon(P.h,Q.h)))<Minimus(variables(Q.h/gcdMon(P.h,
247 Q.h))))
248                    {
249                    d=-1;
250                   //print("Per Indice");
251                    }
252              }
253         }
254  return(d);
255 }
256 //example
257 { "EXAMPLE:"; echo = 2;
258  ring r=0, (x,y,z), rp;
259 jmp p1;
260 p1.h=poly(1);
261 p1.t=poly(1);
262 jmp p2;
263 p2.h=x^2;
264 p2.t=poly(0);
265 jmp p3;
266 p3.h=x;
267 p3.t=poly(0);
268  GJmpMins(p1, p2);
269  GJmpMins(p2, p3);
270 GJmpMins(p1,p1);
271 }
272 ////////////////////////////////////////////////////////////////////////
273  proc TernCompare(list A, list B, list G)
274 /*"USAGE: TernCompare(A,B,C); A list, B list, G list
275 RETURN: int: d
276 NOTE: A and B are terns, while G is the given list of
```

```
277 J−marked polynomials.
278 EXAMPLE: example TernCompare; shows an example"∗/
279 {
280 int d=−1;
281 //Start: A<B
282 if(A[1]==B[1])
283   {
284    if(A[2]==B[2]&& A[3]==B[3])
285      {
286       //print("Uguali");
287       d=0;
288      }
289    else
290      {
291 jmp g1=G[A[2]][A[3]];
292 jmp g2=G[B[2]][B[3]];
293         if(GJmpMins(g1, g2)==1)
294           {
295             //print("Maggiore per il G");
296             d=1;
297           }
298      }
299   }
300 else
301   {
302     if(A[1]>B[1])
303       {
304 //the ordering MUST be rp
305          //print("Maggiore per Lex");
306          d=1;
307       }
308   }
309 return(d);
310 }
311 //example
312 { "EXAMPLE:"; echo = 2;
313  ring r=0, (x,y,z), rp;
314 jmp r1;
315 r1.h=z^3;
```

```
316 r1.t=poly(0);
317 jmp r2;
318 r2.h=z^2*y;
319 r2.t=poly(0);
320 jmp r3;
321 r3.h=z*y^2 ;
322 r3.t=-x^2*y;
323 jmp r4;
324 r4.h=y^5;
325 r4.t=poly(0);
326 list G2F=list(list(r1,r2,r3),list(r4));
327 TernCompare([1,1,1],[x,1,1],G2F);
328 }
329
330 ////////////////////////////////////////////////////////////////////
331  proc MinOfV(list V, list G)
332 /*"USAGE: Minimal(V,G); V list, G list
333 RETURN: int: R
334 NOTE: Input=lista(terne), G.
335 EXAMPLE: example Minimal; shows an example"*/
336 {
337 //Minimal element for a given degree
338 list R=list();
339 list MIN=V[1];
340 int h=1;
341 int i;
342 for(i=2; i<=size(V); i++)
343     {
344 //I consider the first as minimum
345 //If I find something smaller I change minimum
346       if(TernCompare(V[i],MIN,G)<=0)
347         {
348          MIN=V[i];
349          h=i;
350         }
351     }
352 //Return: [minimum,position of the minimum]
353 R=MIN,h;
354 return(R);
```

```
355 }
356 //example
357 { "EXAMPLE:"; echo = 2;
358   ring r=0, (x,y,z), rp;
359 jmp r1;
360 r1.h=z^3;
361 r1.t=poly(0);
362 jmp r2;
363 r2.h=z^2*y;
364 r2.t=poly(0);
365 jmp r3;
366 r3.h=z*y^2 ;
367 r3.t=-x^2*y;
368 jmp r4;
369 r4.h=y^5;
370 r4.t=poly(0);
371 list G2F=list(list(r1,r2,r3),list(r4));
372 MinOfV(VConst(G2F,4,basering)[1],G2F);
373 }
374 ////////////////////////////////////////////////////////////////////////
375   proc OrderingV(list V,list G,list R)
376 /*"USAGE: OrderingV(V,G,R); V list, G list, R list
377 RETURN: list: R
378 NOTE: Input: Vm,G,emptylist
379 EXAMPLE: example OrderingV; shows an example"*/
380 {
381 //Order V[m]
382 //R will contain results but at the beginning it is empty
383 list M=list();
384 if(size(V)==1)
385    {
386       R=insert(R,V[1],size(R));
387    }
388 else
389    {
390     M=MinOfV(V,G);
391     R=insert(R,M[1],size(R));
392     V=delete(V,M[2]);
393 //recursive call
```

```
394     R=OrderingV(V,G,R);
395    }
396 return(R);
397 }
398 example
399 { "EXAMPLE:"; echo = 2;
400  ring r=0, (x,y,z), rp;
401 jmp r1;
402 r1.h=z^3;
403 r1.t=poly(0);
404 jmp r2;
405 r2.h=z^2*y;
406 r2.t=poly(0);
407 jmp r3;
408 r3.h=z*y^2;
409 r3.t=-x^2*y;
410 jmp r4;
411 r4.h=y^5;
412 r4.t=poly(0);
413 list G2F=list(list(r1,r2,r3),list(r4));
414 OrderingV(VConst(G2F,4,basering)[1],G2F,list());
415 }
416 ////////////////////////////////////////////////////////////////////////
417 proc StartOrderingV(list V,list G)
418 /*"USAGE: StartOrdina(V,G); V list, G list
419 RETURN: list: R
420 NOTE: Input Vm,G. This procedure uses OrderingV to get
421 the ordered polynomials as in [BCLR].
422 EXAMPLE: example StartOrderingV; shows an example"*/
423 {
424  return(OrderingV(V,G, list()));
425 }
426 //example
427 { "EXAMPLE:"; echo = 2;
428  ring r=0, (x,y,z), rp;
429 jmp r1;
430 r1.h=z^3;
431 r1.t=poly(0);
432 jmp r2;
```

```
433 r2.h=z^2*y;
434 r2.t=poly(0);
435 jmp r3;
436 r3.h=z*y^2;
437 r3.t=-x^2*y;
438 jmp r4;
439 r4.h=y^5;
440 r4.t=poly(0);
441 list G2F=list(list(r1,r2,r3),list(r4));
442 StartOrderingV(VConst(G2F,4,basering)[1],G2F);
443 }
444 ////////////////////////////////////////////////////////////////////
445  proc Multiply(list L, list G)
446 /*"USAGE: moltiplica(L,G); L list, G list
447 RETURN: jmp: K
448 NOTE: Input: a 3-ple,G. It performs the product associated
449 to the 3-uple.
450 EXAMPLE: example Multiply; shows an example"*/
451 {
452 jmp g=G[L[2]][L[3]];
453 jmp K;
454 K.h=L[1]*g.h;
455 K.t=L[1]*g.t;
456  return(K);
457 }
458 //example
459 { "EXAMPLE:"; echo = 2;
460  ring r=0, (x,y,z), rp;
461  list P=x^2,1,1;
462 jmp r1;
463 r1.h=z^3;
464 r1.t=poly(0);
465 jmp r2;
466 r2.h=z^2*y;
467 r2.t=poly(0);
468 jmp r3;
469 r3.h=z*y^2 ;
470 r3.t=-x^2*y;
471 jmp r4;
```

```
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
Multiply(P,G2F);
}
////////////////////////////////////////////////////////////////////
 proc IdealOfV(list V)
/*"USAGE: IdealOfV(V); V list
RETURN: ideal: I
NOTES: this procedure takes a list of Vm's of a certain degree
and construct their ideal, multiplying the head by the weighted
variable t.
EXAMPLE: example IdealOfV; shows an example"*/
{
ideal I=0;
int i;
if (size(V)!=0)
  {
   list M=list();
jmp g;
   for(i=1; i<= size(V); i++)
       {
          g=V[i];
          g.h=t*g.h;
          M[i]=g.h+g.t;
       }
   I=M[1..size(M)];
//print("IdealOfV");
  //I=std(I);
  }
return(I);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z,t), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
```

```
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
IdealOfV(G2F[1]);
}
////////////////////////////////////////////////////////////////////////
 proc NewWeight(int n)
/*"USAGE: NewWeight(n); n int
RETURN: intvec: u
EXAMPLE: example NewWeight; shows an example"*/
{
intvec u=0;
u[n]=1;
return(u);
}
//example
{ "EXAMPLE:"; echo = 2;
 NewWeight(3);
}
////////////////////////////////////////////////////////////////////////
 proc FinalVm(list V1 , list G1 , r)
/*"USAGE: FinalVm(V1, G1, r); V1 list, G1 list , r
RETURN: intvec: u
EXAMPLE: example NewWeight; shows an example"*/
{
//multiply and reduce, degree by degree
intvec u=NewWeight(nvars(r)+1);
list L=ringlist(r);
L[2]=insert(L[2],"t",size(L[2]));
//print(L[2]);
list ordlist="a",u;
L[3]=insert(L[3],ordlist,0);
def H=ring(L);
```

```
//print(V1);
//print(G1);
list M=list();
jmp p;
list N;
poly q;
poly s;
int i;
int j;
for(i=1; i<=size(G1); i++)
    {
                N=list();
                for(j=1; j<=size(G1[i]); j++)
                    {
                      p=G1[i][j];
                      q=p.h;
                      s=p.t;
                      N[j]=list(q,s);
                     }
                M[i]=N;
     }
p.h=poly(0);
p.t=poly(0);
setring H;
list R=list();
list S=list();
//print("anello definito");
def V=imap(r,V1);
//def G=imap(r,G1);
//print(V);
def MM=imap(r,M);
list G=list();
list N=list();
for(i=1; i<=size(MM); i++)
    {
                for(j=1; j<=size(MM[i]); j++)
                    {
                      p.h=MM[i][j][1];
                      p.t=MM[i][j][2];
```

```
589                     N[j]=p;
590                    }
591                G[i]=N;
592     }
593 ideal I=0;
594 jmp LL;
595 jmp UU;
596  for(i=1; i<=size(V);i++)
597     {
598         R[i]=list();
599         S[i]=list();
600         I=0;
601         for(j=1;j<=size(V[i]); j++)
602             {
603               LL=Multiply(V[i][j],G);
604               LL.t=reduce(t*LL.t,I);
605 //I only reduce the tail
606                  LL.t=subst(LL.t,t,1);
607                  S[i]=insert(S[i],LL,size(S[i]));
608                  LL.h=t*LL.h;
609               R[i]=insert(R[i],LL,size(R[i]));
610                UU=R[i][j];
611                 I=I+ideal(UU.h+UU.t);
612                 attrib(I,"isSB",1);
613             }
614     }
615 list M=list();
616 poly q;
617 poly s;
618 for(i=1; i<=size(S); i++)
619    {
620               N=list();
621               for(j=1; j<=size(S[i]); j++)
622                   {
623                     p=S[i][j];
624                     q=p.h;
625                     s=p.t;
626                     N[j]=list(q,s);
627                    }
```

```
628                 M[i]=N;
629       }
630 p.h=poly(0);
631 p.t=poly(0);
632 setring r;
633 def MM=imap(H,M);
634 list MMM=list();
635 for(i=1; i<=size(MM); i++)
636     {
637                 N=list();
638                 for(j=1; j<=size(MM[i]); j++)
639                     {
640                       p.h=MM[i][j][1];
641                       p.t=MM[i][j][2];
642                       N[j]=p;
643                      }
644                 MMM[i]=N;
645       }
646 return(MMM);
647 }
648 example
649 { "EXAMPLE:"; echo = 2;
650   ring r=0, (x,y,z), rp;
651 jmp r1;
652 r1.h=z^3;
653 r1.t=poly(0);
654 jmp r2;
655 r2.h=z^2*y;
656 r2.t=poly(0);
657 jmp r3;
658 r3.h=z*y^2 ;
659 r3.t=-x^2*y;
660 jmp r4;
661 r4.h=y^5;
662 r4.t=poly(0);
663 list G2F=list(list(r1,r2,r3),list(r4));
664   FinalVm(VConst(G2F,6,r) , G2F, r);
665 }
666 ////////////////////////////////////////////////////////////////////////
```

```
667  proc ConstructorMain(list G, int c,r)
668 /*"USAGE: Costruttore(G,c); G list, c int
669 RETURN: list: R
670 NOTE: At the end separated by degree.
671 EXAMPLE: example Costruttore; shows an example"*/
672 {
673 list V=list();
674 V= VConst(G,c);
675 //print("VConst");
676 //V non ordered
677 list L=list();
678 list R=list();
679 int i;
680 // head, position
681 //order the different degrees
682 for(i=1; i<=size(V); i++)
683     {
684       L[i]=StartOrderingV(V[i], G);
685     }
686 //multiply and reduce
687 //print("Ordinare");
688 R=FinalVm(L, G, r);
689 //print("FinalVm");
690 return(R);
691 }
692 //example
693 { "EXAMPLE:"; echo = 2;
694   ring r=0, (x,y,z), rp;
695 jmp r1;
696 r1.h=z^3;
697 r1.t=poly(0);
698 jmp r2;
699 r2.h=z^2*y;
700 r2.t=poly(0);
701 jmp r3;
702 r3.h=z*y^2 ;
703 r3.t=-x^2*y;
704 jmp r4;
705 r4.h=y^5;
```

```
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
ConstructorMain(G2F,6,r);
}
////////////////////////////////////////////////////////////////////
 proc EKCouples(jmp A, jmp B)
/*"USAGE: CoppiaEK(A,B); A list, B list
RETURN: list: L
NOTE: At the end the monomials involved by EK.
EXAMPLE: example EKCouples; shows an example"*/
{
poly E;
list L=0,0;
string s=varstr(basering);
list VVV=varstr(basering);
//L will contain results
poly h=Minimus(variables(A.h));
//print(h);
int l=findvars(h,1)[2][1];
if(l!=nvars(basering))
 {
//print("vero");
//print(l);
  for(int j=l+1;j<=nvars(basering); j++)
   {
    //print("entrata");
    //print(var(j));
    E=var(j)*A.h/B.h;
//Candidate for * product
    //print(E);
    if(E!=0)
      {
        //print("primo if passato");
        if(Minimus(variables(B.h))>=Maximus(variables(E)))
            {
//Does it work with * ?
              //print("secondo if passato");
              L[1]=j;
              L[2]=E;
```

```
745                   break;
746                 }
747           }
748     }
749   }
750 return (L);
751 }
752 //example
753 { "EXAMPLE:"; echo = 2;
754   ring r=0, (x,y,z), rp;
755 jmp A;
756 A.h=y*z^2;
757 A.t=poly(0);
758 jmp B;
759 B.h=y^2*z;
760 B.t=poly(0);
761 EKCouples(A,B);
762 EKCouples(B,A);
763 }
764 ////////////////////////////////////////////////////////////////////////
765   proc EKPolys(list G)
766 /*"USAGE: PolysEK(G); G list
767 RETURN: list: EK, list: D
768 NOTE: At the end EK polynomials and their degrees
769 EXAMPLE: example PolysEK; shows an example"*/
770 {
771 list D=list();
772 list C=list();
773 list N=0,0;
774 list EK=list();
775 int i;
776 int j;
777 int k;
778 int l;
779 jmp p;
780 for(i=1; i<=size(G); i++)
781     {
782       for(j=1; j<=size(G[i]); j++)
783           {
```

```
784             for(k=1; k<=size(G); k++)
785                 {
786                   for(l=1; l<=size(G[k]); l++)
787                     {
788                       if(i!=k || j!=l)
789                         {
790 //Loop on polynomials
791                           C=EKCouples(G[i][j], G[k][l]);
792 //print("coppia");
793                           if(C[2]!=0)
794                             {
795                               C=insert(C,list(i,j,k,l),size(C));
796                               EK=insert(EK,C,size(EK));
797                               p=G[k][l];
798                               D=insert(D,deg(C[2]*p.h),size(D));
799                             }
800                         }
801                     }
802                 }
803             }
804     }
805 //Double Return
806 return(EK, D);
807 }
808 //example
809 { "EXAMPLE:"; echo = 2;
810    ring r=0, (x,y,z), rp;
811 jmp r1;
812 r1.h=z^3;
813 r1.t=poly(0);
814 jmp r2;
815 r2.h=z^2*y;
816 r2.t=poly(0);
817 jmp r3;
818 r3.h=z*y^2;
819 r3.t=-x^2*y;
820 jmp r4;
821 r4.h=y^5;
822 r4.t=poly(0);
```

```
823 list G2F=list(list(r1,r2,r3),list(r4));
824 EKPolys(G2F);
825 }
826 ////////////////////////////////////////////////////////////////////
827  proc EKPolynomials(list EK, list G)
828 /*"USAGE: EKPolynomials(EK,G); EK list, G list
829 RETURN: list: p
830 NOTE: At the end I obtain the EK polynomials and
831 their degrees.
832 EXAMPLE: example SpolyEK; shows an example"*/
833 {
834 jmp u=G[EK[3][1]][EK[3][2]];
835 jmp q=G[EK[3][3]][EK[3][4]];
836 return(var(EK[1])*(u.h+u.t)-EK[2]*(q.h+q.t));
837 }
838 example
839 { "EXAMPLE:"; echo = 2;
840  ring r=0, (x,y,z), rp;
841 jmp r1;
842 r1.h=z^3;
843 r1.t=poly(0);
844 jmp r2;
845 r2.h=z^2*y;
846 r2.t=poly(0);
847 jmp r3;
848 r3.h=z*y^2;
849 r3.t=-x^2*y;
850 jmp r4;
851 r4.h=y^5;
852 r4.t=poly(0);
853 list G2F=list(list(r1,r2,r3),list(r4));
854 list EK,D=EKPolys(G2F);
855 EKPolynomials(EK[1],G2F);
856 }
857 ////////////////////////////////////////////////////////////////////
858 proc TestJMark(list G1,r)
859 /*"USAGE: TestJMark(G); G list
860 RETURN: int: i
861 NOTE:
```

```
862 This procedure performs J−marked basis test.@∗
863 The input is a list of J−marked polynomials (jmp) arranged
864 by degree, so G1 is a list of list.@∗
865 The output is a boolean evaluation:
866 True=1/False=0
867 EXAMPLE: example TestJMark; shows an example"∗/
868 {int flag=1;
869 if(size(G1)==1 && size(G1[1])==1)
870  {
871   //Hypersurface
872   print("Only␣One␣Polynomial");
873   flag=1;
874  }
875 else
876   {
877    int d=0;
878    list EK,D=EKPolys(G1);
879 //print("PolysEK");
880    //I found EK couples
881    int massimo=Max(D);
882    list V1=ConstructorMain(G1,massimo,r);
883 //print("Costruttore");
884 //print(V1);
885    jmp mi=V1[1][1];
886    int minimo=Min(deg(mi.h));
887    intvec u=NewWeight(nvars(r)+1);
888 list L=ringlist(r);
889 L[2]=insert(L[2],"t",size(L[2]));
890 //print(L[2]);
891 list ordlist="a",u;
892 L[3]=insert(L[3],ordlist,0);
893 def H=ring(L);
894 list JJ=list();
895 jmp pp;
896 jmp qq;
897 int i;
898 int j;
899 list NN;
900 for(i=size(V1);i>0;i−−)
```

```
901     {
902      NN=list();
903      for(j=size(V1[i]);j>0;j--)
904           {
905            //print(j);
906            pp=V1[i][j];
907            NN[j]=list(pp.h,pp.t);
908            }
909 //print(NN);
910 JJ[i]=NN;
911 //print(JJ[i]);
912 //print(i);
913     }
914 //print(JJ);
915 list KK=list();
916 list UU=list();
917 //jmp qq;
918 for(i=size(G1);i>0;i--)
919     {
920      for(j=size(G1[i]);j>0;j--)
921           {
922            //print(j);
923            qq=G1[i][j];
924            UU[j]=list(qq.h,qq.t);
925            }
926 //print(UU);
927 KK[i]=UU;
928     }
929 setring H;
930 //I defined the new ring with the weighted
931 //variable t
932 poly p;
933 //print("anello definito");
934 def JJJ=imap(r,JJ);
935 def EK=imap(r,EK);
936 //print(flag);
937 //imap(r,D);
938 list V=list();
939 jmp fp;
```

```
940 //int i;
941 //int j;
942 list N;
943 for(i=size(JJJ); i>0; i−−)
944     {
945                 N=list();
946                 for(j=size(JJJ[i]); j>0; j−−)
947                     {
948                       fp.h=JJJ[i][j][1];
949                       fp.t=JJJ[i][j][2];
950                       N[j]=fp;
951                      }
952                 V[i]=N;
953                  }
954 //print(V);
955 def KKJ=imap(r,KK);
956 list G=list();
957 list U=list();
958 for(i=1; i<=size(KKJ); i++)
959     {
960                 for(j=1; j<=size(KKJ[i]); j++)
961                     {
962                       fp.h=KKJ[i][j][1];
963                       fp.t=KKJ[i][j][2];
964                       U[j]=fp;
965                      }
966                 G[i]=U;
967      }
968 // print(V);
969 //print(G);
970 //I imported in H everithing I need
971 poly q;
972 ideal I;
973     for(j=1; j<=size(EK);j++)
974         {
975           d=D[j];
976           p=EKPolynomials(EK[j],G);
977           //print("arrivo");
978           I=IdealOfV(V[d−minimo+1]);
```

```
979 attrib(I,"isSB",1);
980 //print(I);
981 q=reduce(t*p,I);
982 //print(I[1]);
983 //print(t*p);
984 q=subst(q,t,1);
985 //I reduce all the EK polynomials
986 // q=RiduzPoly(V[d-minimo+1], p);
987           if(q!=0)
988             {
989 //check whether reduction is 0
990               print("NOT A BASIS");
991             flag=0;
992              break;
993             }
994       }
995   }
996 //print(flag);
997 setring r;
998 //typeof(flag);
999 return(flag);
1000 }
1001 //example
1002 { "EXAMPLE:"; echo = 2;
1003   ring r=0, (x,y,z), rp;
1004 jmp r1;
1005 r1.h=z^3;
1006 r1.t=poly(0);
1007 jmp r2;
1008 r2.h=z^2*y;
1009 r2.t=poly(0);
1010 jmp r3;
1011 r3.h=z*y^2 ;
1012 r3.t=-x^2*y;
1013 jmp r4;
1014 r4.h=y^5;
1015 r4.t=poly(0);
1016 list G2F=list(list(r1,r2,r3),list(r4));
1017 TestJMark(G2F,r);
```

}

A.2 JMBConst.lib: a J-marked schemes constructor.

```
/* ////////////////////////////////////////////////////////////////////////
version="$Id:$";
category="Algebraic Geometry";
// summary description of the library
info="
LIBRARY: JMSConst.lib A library for Singular which constructs J−Marked
Schemes.
AUTHOR: Michela Ceria, email: michela.ceria@unito.it

SEE ALSO: JMBTest_lib

KEYWORDS: J−marked schemes, Borel ideals

OVERVIEW:
The library performs the J−marked computation, as described in [BCLR].
As in JMBTest.lib we construct the V polynomials and we reduce the EK
polynomials w.r.t. them, putting the coefficients as results.


The algorithm terminates only if the ordering is rp.
Anyway, the number of reduction steps is bounded.

REFERENCES:
[CR] Francesca Cioffi, Margherita Roggero,Flat Families by Strongly
Stable Ideals and a Generalization of Groebner Bases,
J. Symbolic Comput. 46, 1070−1084, (2011).@*
[BCLR] Cristina Bertone, Francesca Cioffi, Paolo Lella,
Margherita Roggero, Upgraded methods for the effective
computation of marked schemes on a strongly stable ideal,
Journal of Symbolic Computation
(2012), http://dx.doi.org/10.1016/j.jsc.2012.07.006 @*


SEE ALSO: JMSConst_lib
```

```
36 PROCEDURES:
37 BorelCheck(ideal,r) checks whether the given ideal is Borel
38 JMarkedScheme(list, list, list, int) computes authomatically all the J−marked
39 scheme
40 ";*/
41 LIB "all.lib";
42 ////////////////////////////////////////////////////////////////////////
43 proc BorelCheck(ideal Borid,r)
44 /*"USAGE: BorelCheck(Borid,r); Borid ideal, r ring
45 RETURN: int: d
46 NOTE: Input must be a monomial ideal.
47 The procedure checks whether the Borel moves produce elements belonging to
48 Borid.
49 EXAMPLE: example QuanteC; shows an example"*/
50 {
51 int n= nvars(r);
52 int b=1;
53 int i=1;
54 int k;
55 intvec v;
56 int j;
57 int u;
58 //b =bool. b=1 true; b=0 false
59 //we suppose true!
60 //i=counter on the elements of Borid
61 int s= size(Borid);
62 while(b && i<=s)
63         {
64            v=leadexp(Borid[i]);
65            j=1;
66            u=size(v);
67            while(b && j<=u)
68                  {
69                     if(v[j]!=0)
70                       {
71                         k=j+1;
72                        while(b && k<=n)
73                              {
74                                 b=(reduce(Borid[i]*var(k)/var(j),std(Borid))==0);
```

```
                                    k++;
                                   }
                            }
                     j++;
                    }
          i++;
          }
return(b);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z),rp;
 ideal Borid=y^2*z,y*z^2,z^3,y^5;
BorelCheck(Borid,r);
}
////////////////////////////////////////////////////////////////////////
proc ArrangeBorel(ideal Borid)
/*"USAGE: ArrangeBorel(Borid); Borid ideal
RETURN: list: Input
NOTE: Input must be a monomial ideal, increasingly ordered by degree.
The procedure groups the monomials in a list of lists as needed to compute
J-marked scheme.
// It also returns a list containing the size of every sublist generated.
EXAMPLE: example ArrangeBorel; shows an example"*/
{
list Input;
int j=1;
//list numero=1;
Input[1]=list(Borid[1]);
for(int i=2; i<=size(Borid); i++)
   {
      if(deg(Borid[i])!=deg(Borid[i-1]))
          {
           j++;
           Input[j]=list();
          // numero[j]=0;
          }
Input[j]=insert(Input[j],Borid[i],size(Input[j]));
//numero[j]=numero[j]+1;
```

```
    }
return(Input);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z),rp;
 ideal Borid=y^2*z,y*z^2,z^3,y^5;
ArrangeBorel(Borid);
}
////////////////////////////////////////////////////////////////////////
proc NumNewVar(list B, list NumN)
/*"USAGE: NumNewVar(B,NumN); B list, NumN list
RETURN: int: d
NOTE: B is the grouped Borel, while NumN is a list containing the cardinalities
of the obtained groups.
EXAMPLE: example NumNewVar; shows an example"*/
{
int d;
int j;
int i;
for(i=1; i<=size(B); i++)
    {
      d=d+size(B[i])*NumN[i];
     }
return(d);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z),rp;
 ideal Borid=y^2*z,y*z^2,z^3,y^5;
list B= ArrangeBorel(Borid);
list NumN=7,8;
NumNewVar(B,NumN);
}
////////////////////////////////////////////////////////////////////////
proc NewTails(ideal NI, int s)
/*"USAGE: NewTails(NI,s); NI ideal, s int
RETURN: list: M
NOTE: The procedure construct the tails of the required unknown J−marked
```

```
153 polynomials.
154 EXAMPLE: example NewTails; shows an example"*/
155 {
156  poly p=0;
157 for(int i=1; i<=size(NI); i++)//Loop on the Groebner escalier
158     {
159       p=p+NI[i]*c(i+s); //multiply by c's
160     }
161 int u=size(NI);
162 list M=p,u;
163 return(M);
164 }
165 //example
166 { "EXAMPLE:"; echo = 2;
167  ring r=(0,c(1..7)), (x,y,z),rp;
168  ideal NI=x^2,x*y,y^2,z^2;
169  NewTails(NI,3);
170 }
171 ////////////////////////////////////////////////////////////////////////
172 proc ArrangeTails(list Q)
173 /*"USAGE: ArrangeTails(Q); Q list
174 RETURN: list: Q
175 NOTE: Constructs the final list of J−marked polynomials.
176 EXAMPLE: example FormaInput; shows an example"*/
177 {
178 jmp m=Q[1][1];
179 jmp M=Q[size(Q)][1];
180 int minimo=deg(m.h);
181 int massimo=deg(M.h);
182 //print(minimo);
183 //print(massimo);
184 int i=2;
185 jmp qi;
186 while(i<=size(Q))
187       {
188          //print("entro nel ciclo");
189          //print(i);
190          qi=Q[i][1];
191          if(deg(qi.h)!=minimo+1)
```

```
192             {
193               //print("qui riempire");
194               //print(i);
195               Q=insert(Q,list(),i-1);//Insert empty list for all intermediate degree
196 between the minimum and the maximum, not having polynomials.
197               //print(Q);
198             }
199         minimo=minimo+1;
200         i=i+1;
201         //print("ora ho");
202         //print(minimo);
203         //print(i);
204       }
205 return(Q);
206 }
207 //example
208 { "EXAMPLE:"; echo = 2;
209 ring r=0, (x,y,z),rp;
210  ideal Borid=y^2*z,y*z^2,z^3,y^5;
211 attrib(Borid,"isSB",1);
212     list B=ArrangeBorel(Borid);
213     list NumN;
214     list N;
215     int i;
216     int d;
217     for(i=1;i<=size(B);i++)
218         {
219           d=deg(B[i][1]);
220           N[i]=kbase(Borid,d);
221           NumN[i]=size(N[i]);
222         }
223 int qc=NumNewVar(B, NumN);
224 //Now I must define the NEW RING, putting the c parameters inside.
225 list L=ringlist(r);
226 list L2;
227 L2[1]=L[1];
228 L2[2]=list();
229 for(i=qc;i>=1;i--)
230     {
```

```
231         L2[2][i]="c("+string(i)+")";
232       }
233 L2[3]=list(list("rp",qc));
234 L2[4]=L[4];
235 L[1]=L2;
236 def K=ring(L);
237 setring(K);
238 def Borid=imap(r,Borid);
239 def N=imap(r,N);
240 def B=imap(r,B);
241 //NumN contains only scalars so I do not imap it
242 int j;
243 list Q;
244 int s;
245 list M;
246 jmp pp;
247 for(i=1;i<=size(B);i++)
248       {
249         Q[i]=list();
250        for(j=1;j<=size(B[i]);j++)
251             {
252               M=NewTails(N[i],s);
253               pp.h=B[i][j];
254               pp.t=M[1];
255               Q[i][j]=pp;
256               s=s+M[2];
257               //print(s);
258             }
259       }
260 list P=ArrangeTails(Q);
261 int ll;
262 int uu;
263 jmp Pp;
264 for(ll=1; ll<=size(P);ll++)
265     {
266      for(uu=1;uu<=size(P[ll]);uu++)
267           {Pp=P[ll][uu]; Pp.h; Pp.t;}
268     }
269 }
```

```
270 ////////////////////////////////////////////////////////////////////////
271 proc mod_init()
272 /*"USAGE: mod_init();
273 RETURN: struct: jmp
274 EXAMPLE: example mod_init; shows an example"*/
275 {
276 newstruct("jmp", "poly␣h,␣poly␣t");
277 }
278 //example
279 { "EXAMPLE:"; echo = 2;
280   mod_init();
281 }
282 ////////////////////////////////////////////////////////////////////////
283 proc Terns(list G, int c)
284 /*"USAGE: Terns(G,c); G list, c int
285 RETURN: list: T
286 NOTE: Input is a list of J−marked polynomials
287 (arranged by degree) and an integer.
288 EXAMPLE: example Terns; shows an example"*/
289 {
290 list T=list();
291 int z;
292  for(int k=1; k<=size(G[c]);k=k+1)
293      {
294 //Loop on G[c] making positions of polynomials in G[c]
295         z=size(T);
296         T=insert(T,list(1,c,k) ,size(T));
297      }
298 return(T);
299 }
300 //example
301 { "EXAMPLE:"; echo = 2;
302 ring r=0, (x,y,z), rp;
303 jmp r1;
304 r1.h=z^3;
305 r1.t=poly(0);
306 jmp r2;
307 r2.h=z^2*y;
308 r2.t=poly(0);
```

```
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
Terns(G2F, 1);
Terns(G2F, 2);
}
////////////////////////////////////////////////////////////////
proc VConst(list G, int c)
/*"USAGE: VConst(G, c); G list, c int
RETURN: list: V
NOTES: this procedure computes the Vm polynomials following the
algorithm in [CR],but it only keeps in memory the monomials by
which the G's must be multplied and their positions.
EXAMPLE: example VConst; shows an example"*/
{
jmp f=G[1][1];
int aJ=deg(f.h);
// minimal degree of polynomials in G
//print(aJ);
list V=list();
V[1]=Terns(G,1);
// V[1]=G[1] (keeping in memory only [head, position])
//print(c-aJ+1);
int i;
int j;
int m;
list OO;
jmp p;
  for(m=2; m<=c-aJ+1; m=m+1)
  {
      //print("entro nel form");
      if(m>size(G))
          {V[m]=list();
//If we have not G[m] we insert a list()
          //print("vuota prima");
```

```
348          }
349       else
350          {V[m]=Terns(G,m);
351          //print("piena prima");
352          }
353       for(i=1; i<nvars(basering)+1; i=i+1)
354          {
355            //print("entrata fori");
356            //print(i);
357            for(j=1; j<=size(V[m−1]); j=j+1)
358               {
359                     p=G[V[m−1][j][2]][V[m−1][j][3]];
360                    //print(p.h);
361                    //print(p.t);
362                    //print(var(i));
363                    //print(Minimus(V[m−1][j][1]∗p.h));
364                  if(var(i)<=Minimus(variables(V[m−1][j][1]∗p.h)))
365                    {
366 //Can I multiply by the current variable?
367                      //print("minoremin");
368                      //print("fin qui ci sono");
369                       //print(V[m−1][j][1]);
370                       OO=list(var(i)∗V[m−1][j][1],V[m−1][j][2],V[m−1][j][3]);
371                      V[m]=insert(V[m], OO ,size(V[m]));
372                    }
373               }
374          }
375    }
376 return (V);}
377 //example
378 { "EXAMPLE:"; echo = 2;
379   ring r=0, (x,y,z), rp;
380 jmp r1;
381 r1.h=z^3;
382 r1.t=poly(0);
383 jmp r2;
384 r2.h=z^2∗y;
385 r2.t=poly(0);
386 jmp r3;
```

```
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
VConst(G2F,4,basering);}
////////////////////////////////////////////////////////////////////
proc Minimus(ideal L)
/*"USAGE: Minimus(L); G list, c int
RETURN: list: V
NOTES: it returns the minimal variable generating the ideal L;
input must be an ideal generated by variables.
EXAMPLE: example Minimus; shows an example"*/
{
poly min=L[1];
int i;
for(i=2;i<=size(L); i++)
  {
    if(L[i]<min){min=L[i];}
  }
return(min);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z), rp;
ideal I=y,x,z;
Minimus(I);
}
////////////////////////////////////////////////////////////////////
proc Maximus(ideal L)
/*"USAGE: Maximus(L); G list, c int
RETURN: list: V
NOTES: it returns the maximal variable generating the ideal L
input must be an ideal generated by variables.
EXAMPLE: example Maximus; shows an example"*/
{
poly max=L[1];
int i;
```

```
for(i=2;i<=size(L); i++)
  {
    if(L[i]>max){max=L[i];}
  }
return(max);
}
example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z), rp;
ideal I=y,x,z;
Maximus(I);
}
////////////////////////////////////////////////////////////////////////
proc GPolyMin(jmp P, jmp Q)
/*"USAGE: GPolyMin(P,Q); P jmp, Q jmp
RETURN: int: d
EXAMPLE: example GPolyMin; shows an example"*/
{
  int d=1;
//−1=lower, 0=equal, 1=higher
//At the beginning suppose Q is higher
  if(deg(P.h)<deg(Q.h))
      {
//Compare degrees;
       d=−1;
       //print("Per Grado");
      }
  if(deg(P.h)==deg(Q.h))
      {
       if(P.h==Q.h)
           {
            if(P.t==Q.t)
                {
//head=tail
                 d=0;
                 //print("Uguali");
                }
           }
      else
```

```
465             {
466 //print(Minimus(variables(P.h/gcdMon(P.h,Q.h))));
467 //print(Minimus(variables(Q.h/gcdMon(P.h,Q.h))));
468
469 if(Minimus(variables(P.h/gcdMon(P.h,Q.h)))<Minimus(variables(Q.h/gcdMon(P.h,
470 Q.h))))
471                     {
472                     d=-1;
473                    //print("Per Indice");
474                     }
475             }
476         }
477  return(d);
478 }
479 //example
480 { "EXAMPLE:"; echo = 2;
481  ring r=0, (x,y,z), rp;
482  jmp p1;
483 p1.h=poly(1);
484 p1.t=poly(1);
485 jmp p2;
486 p2.h=x^2;
487 p2.t=poly(0);
488 jmp p3;
489 p3.h=x;
490 p3.t=poly(0);
491  GPolyMin(p1,p2);
492  GPolyMin(p2, p3);
493 GPolyMin(p2,p2);
494 }
495 ////////////////////////////////////////////////////////////////////
496 proc TernComparer(list A, list B, list G)
497 /*"USAGE: TernComparer(A,B,C); A list, B list, G list
498 RETURN: int: d
499 NOTE: A and B are terns, while G is the given list of
500 J-marked polynomials.
501 EXAMPLE: example TernComparer; shows an example"*/
502 {
503 int d=-1;
```

```
504 //Start: A<B
505 if(A[1]==B[1])
506   {
507    if(A[2]==B[2]&& A[3]==B[3])
508      {
509       //print("Uguali");
510       d=0;
511      }
512    else
513      {
514 jmp g1=G[A[2]][A[3]];
515 jmp g2=G[B[2]][B[3]];
516         if(GPolyMin(g1, g2)==1)
517           {
518             //print("Maggiore per il G");
519             d=1;
520           }
521      }
522   }
523 else
524   {
525     if(A[1]>B[1])
526       {
527 //the ordering MUST be rp
528          //print("Maggiore per Lex");
529          d=1;
530       }
531   }
532 return(d);
533 }
534 //example
535 { "EXAMPLE:"; echo = 2;
536  ring r=0, (x,y,z), rp;
537 jmp r1;
538 r1.h=z^3;
539 r1.t=poly(0);
540 jmp r2;
541 r2.h=z^2*y;
542 r2.t=poly(0);
```

```
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
TernComparer([1,1,1],[x,1,1],G2F);
}
////////////////////////////////////////////////////////////////////////
proc MinimalV(list V, list G)
/*"USAGE: Minimal(V,G); V list, G list
RETURN: int: R
NOTE: Input=list(terns), G.
EXAMPLE: example MinimalV; shows an example"*/
{
//Minimal element for a given degree
list R=list();
list MIN=V[1];
int h=1;
int i;
for(i=2; i<=size(V); i++)
      {
//I consider the first as minimum
//If I find something smaller I change minimum
       if(TernComparer(V[i],MIN,G)<=0)
          {
           MIN=V[i];
           h=i;
          }
      }
//Return: [minimum,position of the minimum]
R=MIN,h;
return(R);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z), rp;
jmp r1;
```

```
582 r1.h=z^3;
583 r1.t=poly(0);
584 jmp r2;
585 r2.h=z^2*y;
586 r2.t=poly(0);
587 jmp r3;
588 r3.h=z*y^2 ;
589 r3.t=-x^2*y;
590 jmp r4;
591 r4.h=y^5;
592 r4.t=poly(0);
593 list G2F=list(list(r1,r2,r3),list(r4));
594 MinimalV(VConst(G2F,4,basering)[1],G2F);
595 }
596 ////////////////////////////////////////////////////////////////////////
597 proc OrderV(list V,list G,list R)
598 /*"USAGE: Ordinare(V,G,R); V list, G list, R list
599 RETURN: list: R
600 NOTE: Input: Vm,G,emptylist
601 EXAMPLE: example Ordinare; shows an example"*/
602 {
603 //Order V[m]
604 //R will contain results but at the beginning it is empty
605 list M=list();
606 if(size(V)==1)
607   {
608     R=insert(R,V[1],size(R));
609   }
610 else
611   {
612    M=MinimalV(V,G);
613    R=insert(R,M[1],size(R));
614    V=delete(V,M[2]);
615 //recursive call
616    R=OrderV(V,G,R);
617   }
618 return(R);
619 }
620 //example
```

```
{ "EXAMPLE:"; echo = 2;
  ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
OrderV(VConst(G2F,4,r)[1],G2F,list());
}
////////////////////////////////////////////////////////////////////////
proc StartOrderingV(list V,list G)
/*"USAGE: StartOrderingV(V,G); V list, G list
RETURN: list: R
NOTE: Input Vm,G. This procedure uses OrderV to get
the ordered polynomials as in [BCLR].
EXAMPLE: example StartOrderingV; shows an example"*/
{
  return(OrderV(V,G, list()));
}
//example
{ "EXAMPLE:"; echo = 2;
  ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2;
r3.t=-x^2*y;
```

```
660 jmp r4;
661 r4.h=y^5;
662 r4.t=poly(0);
663 list G2F=list(list(r1,r2,r3),list(r4));
664 StartOrderingV(VConst(G2F,4,basering)[1],G2F);
665 }
666 ////////////////////////////////////////////////////////////////////
667 proc MultiplyJmP(list L, list G)
668 /*"USAGE: MultiplyJmP(L,G); L list, G list
669 RETURN: jmp: K
670 NOTE: Input: a 3−ple,G. It performs the product associated
671 to the 3−uple.
672 EXAMPLE: example MultiplyJmP; shows an example"*/
673 {
674 jmp g=G[L[2]][L[3]];
675 jmp K;
676 K.h=L[1]*g.h;
677 K.t=L[1]*g.t;
678  return(K);
679 }
680 //example
681 { "EXAMPLE:"; echo = 2;
682  ring r=0, (x,y,z), rp;
683  list P=x^2,1,1;
684 jmp r1;
685 r1.h=z^3;
686 r1.t=poly(0);
687 jmp r2;
688 r2.h=z^2*y;
689 r2.t=poly(0);
690 jmp r3;
691 r3.h=z*y^2 ;
692 r3.t=−x^2*y;
693 jmp r4;
694 r4.h=y^5;
695 r4.t=poly(0);
696 list G2F=list(list(r1,r2,r3),list(r4));
697 MultiplyJmP(P,G2F);
698 }
```

```
699 ////////////////////////////////////////////////////////////////////////
700 //proc JmpIdeal(list V,r)
701 //"USAGE: JmpIdeal(V); V list
702 //RETURN: ideal: I
703 //NOTES: this procedure takes a list of Vm's of a certain degree
704 //and construct their ideal, multiplying the head by the weighted
705 //variable t.
706 //EXAMPLE: example JmpIdeal; shows an example"
707 //{
708 //ideal I=0;
709 //int i;
710 //if (size(V)!=0)
711  // {
712 // list M=list();
713 //jmp g;
714 // for(i=1; i<= size(V); i++)
715 // {
716 // g=V[i];
717 // g.h=(g.h)*t;
718 // M[i]=g.h+g.t;
719 // }
720 // I=M[1..size(M)];
721 //attrib(I,"isSB",1);
722 // }
723 //return(I);
724 //}
725 //example
726 //{ "EXAMPLE:"; echo = 2;
727 // ring r=0, (x,y,z,t), rp;
728 //jmp r1;
729 //r1.h=z^3;
730 //r1.t=poly(0);
731 //jmp r2;
732 //r2.h=z^2*y;
733 //r2.t=poly(0);
734 //jmp r3;
735 //r3.h=z*y^2 ;
736 //r3.t=-x^2*y;
737 //jmp r4;
```

```
738 //r4.h=y^5;
739 //r4.t=poly(0);
740 //list G2F=list(list(r1,r2,r3),list(r4));
741 //JmpIdeal(VConst(G2F,6,r)[1],r);
742 //}
743 ////////////////////////////////////////////////////////////////////////
744 proc NewWeight(int n)
745 /*"USAGE: NewWeight(n); n int
746 RETURN: intvec: u
747 EXAMPLE: example NewWeight; shows an example"*/
748 {
749 intvec u=0;
750 u[n]=1;
751 return(u);
752 }
753 //example
754 { "EXAMPLE:"; echo = 2;
755   NewWeight(3);
756 }
757 ////////////////////////////////////////////////////////////////////////
758 proc FinalVm(list V1 , list G1 , r)
759 /*"USAGE: FinalVm(V1, G1, r); V1 list, G1 list , r
760 RETURN: intvec: u
761 EXAMPLE: example FinalVm; shows an example"*/
762 {
763 //multiply and reduce, degree by degree
764 intvec u=NewWeight(nvars(r)+1);
765 list L=ringlist(r);
766 L[2]=insert(L[2],"t",size(L[2]));
767 //print(L[2]);
768 list ordlist="a",u;
769 L[3]=insert(L[3],ordlist,0);
770 def H=ring(L);
771 //print(V1);
772 //print(G1);
773 list M=list();
774 jmp p;
775 list N;
776 poly q;
```

```
poly s;
int i;
int j;
for(i=1; i<=size(G1); i++)
    {
                N=list();
                for(j=1; j<=size(G1[i]); j++)
                    {
                      p=G1[i][j];
                      q=p.h;
                      s=p.t;
                      N[j]=list(q,s);
                     }
                M[i]=N;
     }
//print("M is");
//print(M);
p.h=poly(0);
p.t=poly(0);
setring H;
list R=list();
list S=list();
//print("anello definito");
def V=imap(r,V1);
//def G=imap(r,G1);
//print(V);
def MM=imap(r,M);
list G=list();
list N=list();
for(i=1; i<=size(MM); i++)
    {
                for(j=1; j<=size(MM[i]); j++)
                    {
                      p.h=MM[i][j][1];
                      p.t=MM[i][j][2];
                      N[j]=p;
                     }
                G[i]=N;
     }
```

```
816 ideal I=0;
817 jmp LL;
818 jmp UU;
819 //print("pronta x ridurre");
820  for(i=1; i<=size(V);i++)
821     {
822 //print("sono a V di");
823 //print(i);
824         R[i]=list();
825         S[i]=list();
826         I=0;
827                 attrib(I,"isSB",1);
828         for(j=1;j<=size(V[i]); j++)
829            {
830 //print(j);
831 //print("esimo elem");
832               LL=MultiplyJmP(V[i][j],G);
833                 LL.t=reduce(t*LL.t,I);
834 //I only reduce the tail
835 //print(LL.t);
836                 LL.t=subst(LL.t,t,1);
837                 S[i]=insert(S[i],LL,size(S[i]));
838                 LL.h=t*LL.h;
839               R[i]=insert(R[i],LL,size(R[i]));
840                UU=R[i][j];
841                 I=I+ideal(UU.h+UU.t);
842                 attrib(I,"isSB",1);
843            }
844     }
845 //print("ho ridotto");
846 list M=list();
847 poly q;
848 poly s;
849 for(i=1; i<=size(S); i++)
850    {
851              N=list();
852              for(j=1; j<=size(S[i]); j++)
853                 {
854                    p=S[i][j];
```

```
                    q=p.h;
                    s=p.t;
                    N[j]=list(q,s);
                   }
              M[i]=N;
    }
p.h=poly(0);
p.t=poly(0);
setring r;
def MM=imap(H,M);
list MMM=list();
for(i=1; i<=size(MM); i++)
   {
              N=list();
              for(j=1; j<=size(MM[i]); j++)
                  {
                    p.h=MM[i][j][1];
                    p.t=MM[i][j][2];
                    N[j]=p;
                   }
              MMM[i]=N;
    }
return(MMM);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2 ;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
```

```
894 list G2F=list(list(r1,r2,r3),list(r4));
895  FinalVm(VConst(G2F,6,r) , G2F, r);
896 }
897 ////////////////////////////////////////////////////////////////////
898 proc VmConstructor(list G, int c,r)
899 /*"USAGE: VmConstructor(G,c); G list, c int
900 RETURN: list: R
901 NOTE: At the end separated by degree.
902 EXAMPLE: example VmConstructor; shows an example"*/
903 {
904 list V=list();
905 V= VConst(G,c);
906 //print("VConst");
907 //V non ordered
908 list L=list();
909 list R=list();
910 int i;
911 // head, position
912 //order the different degrees
913 for(i=1; i<=size(V); i++)
914     {
915      L[i]=StartOrderingV(V[i], G);
916     }
917 //print("finito ordine");
918 //multiply and reduce
919 //print("Ordinare");
920 //R=FinalVm(L, G, r);
921 //print("FinalVm");
922 return(L);
923 }
924 //example
925 { "EXAMPLE:"; echo = 2;
926  ring r=0, (x,y,z), rp;
927 jmp r1;
928 r1.h=z^3;
929 r1.t=poly(0);
930 jmp r2;
931 r2.h=z^2*y;
932 r2.t=poly(0);
```

```
933 jmp r3;
934 r3.h=z*y^2 ;
935 r3.t=-x^2*y;
936 jmp r4;
937 r4.h=y^5;
938 r4.t=poly(0);
939 list G2F=list(list(r1,r2,r3),list(r4));
940  VmConstructor(G2F,6,r);
941 }
942 ////////////////////////////////////////////////////////////////////////
943 proc EKCouples(jmp A, jmp B)
944 /*"USAGE: CoppiaEK(A,B); A list, B list
945 RETURN: list: L
946 NOTE: At the end the monomials involved by EK.
947 EXAMPLE: example EKCouples; shows an example"*/
948 {
949 poly E;
950 list L=0,0;
951 string s=varstr(basering);
952 list VVV=varstr(basering);
953 //L will contain results
954 poly h=Minimus(variables(A.h));
955 //print(h);
956 int l=findvars(h,1)[2][1];
957 if(l!=nvars(basering))
958  {
959 //print("vero");
960 //print(l);
961   for(int j=l+1;j<=nvars(basering); j++)
962    {
963     //print("entrata");
964     //print(var(j));
965     E=var(j)*A.h/B.h;
966 //Candidate for * product
967     //print(E);
968     if(E!=0)
969       {
970         //print("primo if passato");
971         if(Minimus(variables(B.h))>=Maximus(variables(E)))
```

```
                {
//Does it work with * ?
                  //print("secondo if passato");
                  L[1]=j;
                  L[2]=E;
                  break;
                }
          }
     }
   }
return (L);
}
//example
{ "EXAMPLE:"; echo = 2;
  ring r=0, (x,y,z), rp;
jmp A;
A.h=y*z^2;
A.t=poly(0);
jmp B;
B.h=y^2*z;
B.t=poly(0);
EKCouples(A,B);
EKCouples(B,A);
}
////////////////////////////////////////////////////////////////////////
proc EKPolynomials(list G)
/*"USAGE: EKPolynomials(G); G list
RETURN: list: EK, list: D
NOTE: At the end EK polynomials and their degrees

EXAMPLE: example EKPolynomials; shows an example"*/
{
list D=list();
list C=list();
list N=0,0;
list EK=list();
int i;
int j;
int k;
```

```
int l;
jmp p;
for(i=1; i<=size(G); i++)
   {
      for(j=1; j<=size(G[i]); j++)
          {
            for(k=1; k<=size(G); k++)
                {
                   for(l=1; l<=size(G[k]); l++)
                       {
                          if(i!=k || j!=l)
                            {
//Loop on polynomials
                             C=EKCouples(G[i][j], G[k][l]);
//print("coppia");
                             if(C[2]!=0)
                               {
                                 C=insert(C,list(i,j,k,l),size(C));
                                 EK=insert(EK,C,size(EK));
                                 p=G[k][l];
                                 D=insert(D,deg(C[2]*p.h),size(D));
                               }
                            }
                       }
                }
          }
   }
//Double Return
return(EK, D);
}
//example
{ "EXAMPLE:"; echo = 2;
   ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
```

```
jmp r3;
r3.h=z*y^2;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
list G2F=list(list(r1,r2,r3),list(r4));
EKPolynomials(G2F);
}
////////////////////////////////////////////////////////////////////////
proc MultEKPolys(list EK, list G)
/*"USAGE: MultEKPolys(G); G list
RETURN: list: p
NOTE: At the end I obtain the EK polynomials and
their degrees.
EXAMPLE: example MultEKPolys; shows an example"*/
{
jmp u;
u=G[EK[3][1]][EK[3][2]];
//print("u");
jmp q;
q=G[EK[3][3]][EK[3][4]];
return(var(EK[1])*(u.h+u.t)-EK[2]*(q.h+q.t));
}
//example
{ "EXAMPLE:"; echo = 2;
  ring r=0, (x,y,z), rp;
jmp r1;
r1.h=z^3;
r1.t=poly(0);
jmp r2;
r2.h=z^2*y;
r2.t=poly(0);
jmp r3;
r3.h=z*y^2;
r3.t=-x^2*y;
jmp r4;
r4.h=y^5;
r4.t=poly(0);
```

```
1089 list G2F=list(list(r1,r2,r3),list(r4));
1090 list EK,D=EKPolynomials(G2F);
1091 MultEKPolys(EK[2],G2F);
1092 }
1093 ////////////////////////////////////////////////////////////////////
1094 proc SchemeEq(list W, list EK,list D,list Q,r)
1095 /*"USAGE: SchemeEq(W,EK,D,Q,r); W list, EK list, D list, Q list, r ring
1096 RETURN: int: i
1097 NOTE:
1098 This procedure performs the reduction of EK−polynomials, obtaining
1099 the J−marked scheme.
1100 EXAMPLE: example SchemeEq; shows an example"*/
1101 {
1102 list Jms=list();
1103 //ideal I;
1104 list M=list();
1105 jmp mini;
1106 mini=W[1][1];
1107 int minimo=deg(mini.h);
1108 //multiply variables
1109 poly pd=poly(1);
1110 for(int i=1;i<=nvars(r);i++)
1111 {pd=pd*var(i);}
1112 //CHANGE RING
1113 intvec u=NewWeight(nvars(r)+1);
1114 list L=ringlist(r);
1115 L[2]=insert(L[2],"t",size(L[2]));
1116 //print(L[2]);
1117 list ordlist="a",u;
1118 L[3]=insert(L[3],ordlist,0);
1119 def H=ring(L);
1120 //list
1121 M=list();
1122 jmp pu;
1123 list N;
1124 poly q;
1125 poly s;
1126 i=0;
1127 int j;
```

```
for(i=1; i<=size(Q); i++)
    {
                N=list();
                for(j=1; j<=size(Q[i]); j++)
                    {
                      pu=Q[i][j];
                      q=pu.h;
                      s=pu.t;
                      N[j]=list(q,s);
                     }
                M[i]=N;
     }
list O;
pu.h=poly(0);
pu.t=poly(0);
for(i=1; i<=size(W); i++)
    {
                N=list();
                for(j=1; j<=size(W[i]); j++)
                    {
                      pu=W[i][j];
                      q=pu.h;
                      s=pu.t;
                      N[j]=list(q,s);
                     }
                O[i]=N;
     }
pu.h=poly(0);
pu.t=poly(0);
setring H;
list R=list();
list S=list();
//print("anello definito");
def EK=imap(r,EK);
def MM=imap(r,M);
def OO=imap(r,O);
def pd=imap(r,pd);
list G=list();
list N=list();
```

```
for(i=1; i<=size(MM); i++)
    {
                for(j=1; j<=size(MM[i]); j++)
                    {
                      pu.h=MM[i][j][1];
                      pu.t=MM[i][j][2];
                      N[j]=pu;
                     }
                G[i]=N;
     }
list V;
for(i=1; i<=size(OO); i++)
    {
                for(j=1; j<=size(OO[i]); j++)
                    {
                      pu.h=OO[i][j][1];
                      pu.t=OO[i][j][2];
                      N[j]=pu;
                     }
                V[i]=N;
     }
//print(V);
//print(G);
matrix C;
list COEFF;
poly p=0;
poly q=0;
ideal I;
list M;
i=0;
jmp g;
int k;
for(j=1; j<=size(EK);j++)
        {
          //print("arrivo");
          //print(j);
          p=MultEKPolys(EK[j],G);
          //ideal
           I=0;
```

```
1206          if (size(V[D[j]−minimo+1])!=0)
1207              {
1208                M=list();
1209              // jmp g;
1210          for(i=1; i<= size(V[D[j]−minimo+1]); i++)
1211              {
1212                g=V[D[j]−minimo+1][i];
1213                g.h=(g.h)∗t;
1214                M[i]=g.h+g.t;
1215              }
1216         I=M[1..size(M)];
1217        attrib(I,"isSB",1);
1218 //print(I);
1219    }
1220 //print(I);
1221 q=reduce(t∗p,I);
1222 q=subst(q,t,1);
1223          C=coef(q,pd);
1224          COEFF=C[2,1..ncols(C)];
1225          for(k=1;k<=size(COEFF);k++)
1226              {
1227                if(COEFF[k]!=0)
1228               { Jms=insert(Jms,COEFF[k],size(Jms));}
1229              }
1230         }
1231 setring r;
1232 def Jms=imap(H,Jms);
1233 return(Jms);
1234 }
1235 //example
1236 { "EXAMPLE:"; echo = 2;
1237   ring r=0, (x,y,z),rp;
1238   ideal Borid=y^2∗z,y∗z^2,z^3,y^5;
1239 attrib(Borid,"isSB",1);
1240       list B=ArrangeBorel(Borid);
1241       list NumN;
1242       list N;
1243       int i;
1244       int d;
```

```
1245      for(i=1;i<=size(B);i++)
1246          {
1247           d=deg(B[i][1]);
1248           N[i]=kbase(Borid,d);
1249           NumN[i]=size(N[i]);
1250          }
1251 int qc=NumNewVar(B, NumN);
1252 //Now I must define the NEW RING,
1253 //putting the c parameters inside.
1254 list L=ringlist(r);
1255 list L2;
1256 L2[1]=L[1];
1257 L2[2]=list();
1258 for(i=qc;i>=1;i--)
1259      {
1260       L2[2][i]="c("+string(i)+")";
1261      }
1262 L2[3]=list(list("rp",qc));
1263 L2[4]=L[4];
1264 L[1]=L2;
1265 if(defined(K)){kill K;}
1266 def K=ring(L);
1267 export K;
1268 setring(K);
1269 def Borid=imap(r,Borid);
1270 def N=imap(r,N);
1271 def B=imap(r,B);
1272 //NumN contains only scalars so I do not imap it
1273 int j;
1274 list Q;
1275 int s;
1276 list M;
1277 jmp pp;
1278 for(i=1;i<=size(B);i++)
1279      {
1280        Q[i]=list();
1281       for(j=1;j<=size(B[i]);j++)
1282           {
1283              M=NewTails(N[i],s);
```

```
1284              pp.h=B[i][j];
1285              pp.t=M[1];
1286              Q[i][j]=pp;
1287              s=s+M[2];
1288              //print(s);
1289            }
1290      }
1291 list P=ArrangeTails(Q);
1292 list EK,D= EKPolynomials(P);
1293       int massimo=Max(D);
1294 //list V=VConst(P, massimo);
1295 //pause();
1296 list V=VmConstructor(P,massimo,r);
1297 list W=FinalVm(V,P,K);
1298 //print("I V ridotti in ordine sono");
1299 //print(W);
1300 list Jms=SchemeEq(W,EK,D,P,K);
1301 Jms;}
1302
1303 ////////////////////////////////////////////////////////////////////////
1304 proc JMarkedScheme(ideal Borid,r)
1305 /*"USAGE: JMarkedScheme(Borid, r); Borid ideal, r ring
1306 RETURN: list: Jms
1307 NOTE:
1308 This procedure performs automatically the whole construction
1309 of the J−marked scheme.
1310 EXAMPLE: example JMarkedScheme; shows an example"*/
1311 {
1312 list Jms;
1313 if(BorelCheck(Borid,r))
1314   {
1315    if(size(Borid)==1)
1316      { Jms=list();}
1317   else{
1318     //print("Input is OK");
1319     attrib(Borid,"isSB",1);
1320     list B=ArrangeBorel(Borid);
1321     list NumN;
1322     list N;
```

```
1323      int i;
1324      int d;
1325      for(i=1;i<=size(B);i++)
1326          {
1327           d=deg(B[i][1]);
1328           N[i]=kbase(Borid,d);
1329           NumN[i]=size(N[i]);
1330          }
1331 int qc=NumNewVar(B, NumN);
1332 if(qc==0)
1333 {Jms=list(0);}
1334 else
1335  {
1336 //Now I must define the NEW RING,
1337 //putting the c parameters inside.
1338 list L=ringlist(r);
1339 list L2;
1340 L2[1]=L[1];
1341 L2[2]=list();
1342 for(i=qc;i>=1;i--)
1343      {
1344       L2[2][i]="c("+string(i)+")";
1345      }
1346 L2[3]=list(list("rp",qc));
1347 L2[4]=L[4];
1348 L[1]=L2;
1349 if(defined(K)){kill K;}
1350 def K=ring(L);
1351 export K;
1352 setring(K);
1353 def Borid=imap(r,Borid);
1354 def N=imap(r,N);
1355 def B=imap(r,B);
1356 //NumN contains only scalars so I do not imap it
1357 int j;
1358 list Q;
1359 int s;
1360 list M;
1361 jmp pp;
```

```
for(i=1;i<=size(B);i++)
    {
      Q[i]=list();
     for(j=1;j<=size(B[i]);j++)
        {
          M=NewTails(N[i],s);
          pp.h=B[i][j];
          pp.t=M[1];
          Q[i][j]=pp;
          s=s+M[2];
          //print(s);
        }
    }
list P=ArrangeTails(Q);
list EK,D= EKPolynomials(P);
      int massimo=Max(D);
//list V=VConst(P, massimo);
//pause();
list V=VmConstructor(P,massimo,r);
list W=FinalVm(V,P,K);
//print("I V ridotti in ordine sono");
//print(W);
//list
Jms=SchemeEq(W,EK,D,P,K);
keepring K;}
}
  }
else
   {
    print("WRONG IDEAL IN INPUT");
    print("It is NOT BOREL");
   }
return(Jms);
}
//example
{ "EXAMPLE:"; echo = 2;
 ring r=0, (x,y,z),rp;
 ideal Borid=y^2*z,y*z^2,z^3,y^5;
JMarkedScheme(Borid,r);}
```

APPENDIX **B**

Locator polynomials and points structures for $\mathbb{F}_8, \mathbb{F}_{16}$

B.1 Cyclical configurations in $\mathbb{F}_8$.

B.1.1 The seven cyclical configurations.

We display here all the data concerning the seven cyclical configurations defined in section 8.3.
All the polynomials have been computed using Singular.

Configuration 1

Number of points	Third coordinate
7	a^2
6	a^3
5	a^4
4	a^5
3	a^6
2	1
1	a

The points are:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a^2, a)],$$
$$[(1, a^5, a^3, a)],$$
$$[(a^2, a^2, a^4, a)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$
$$[(a^3, a^5, a^2, a^5)],$$

$$[(1, a^3, a^2, a^6)],$$
$$[(a^6, a^2, a^2, 1)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, a^3, 1)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, a^4, 1)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

The tower structure is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^4	a^6,a^3,a^3	1,a^6,a^4
a,a^6,a^3	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,a^2	1,a^3,a^2
a,a,a^2	a^2,a^2,a^4	a^3,a,1	a^4,a^4,a^2	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a^3
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

while the configuration list is

$$[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

and the associated polynomial is

$$z_1 + a^6x_1^5x_2^3 + a^5x_1^4x_2^3 + a^5x_1^3x_2^3 + a^3x_1^2x_2^3 +$$
$$a^6x_1x_2^3 + a^3x_2^3 + a^5x_1^6x_2^2 + a^3x_1^5x_2^2 + a^6x_1^4x_2^2 +$$
$$a^3x_1^3x_2^2 + a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^5x_1^3x_2 + a^3x_1x_2$$
$$+a^3x_1^4 + ax_1 + a^6.$$

Configuration 2

Number of points	Third coordinate
7	a^3
6	a^4
5	a^5
4	a^6
3	1
2	a
1	a^2

The points are:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a^3, a)],$$
$$[(a^2, a^2, a^4, a)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^3, a^2)],$$
$$[(a, a, a^4, a^2)],$$
$$[(a^3, a^5, a^5, a^2)],$$
$$[(1, a^3, a^6, a^2)],$$

$$[(a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, a^3, 1)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, a^4, 1)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

The tower structure is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^4	a^6,a^3,a^3	1,a^6,a^4
a,a^6,a^3	a^2,a^4,a^3	a^3,a^5,a^5	a^4,a,a^3	a^5,1,a^3	a^6,a^2,1	1,a^3,a^6
a,a,a^4	a^2,a^2,a^4	a^3,a,1	a^4,a^4,a	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a^3
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

corresponding to

$$[2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

and to

$$z_1 + x_1^5x_2^3 + x_1^4x_2^3 + ax_1^3x_2^3 + x_1^2x_2^3 + a^4x_1x_2^3 +$$
$$a^2x_2^3 + ax_1^6x_2^2 + x_1^5x_2^2 + a^4x_1^4x_2^2 + a^2x_1^3x_2^2 +$$
$$a^6x_1^2x_2^2 + a^4x_1^5x_2 + x_1^3x_2 + x_1x_2 + x_1^4 + ax_1 + 1$$

Configuration 3

Number of points	Third coordinate
7	a^4
6	a^5
5	a^6
4	1
3	a
2	a^2
1	a^3

The points are:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a^4, a)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^4, a^2)],$$
$$[(a^3, a^5, a^5, a^2)],$$
$$[(1, a^3, a^6, a^2)],$$
$$[(a^6, a^2, 1, a^2)],$$
$$[a^6, a^3, a^4, a^3)],$$
$$[(a^2, a^4, a^5, a^3)],$$
$$[(a^4, a, a^6, a^3)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, a^4, 1)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

The point configuration is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^6	a^6,a^3,a^4	1,a^6,a^5
a,a^6,1	a^2,a^4,a^5	a^3,a^5,a^5	a^4,a,a^6	a^5,1,a^2	a^6,a^2,1	1,a^3,a^6
a,a,a^4	a^2,a^2,a^4	a^3,a,1	a^4,a^4,a	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

and the corresponding list is

$$[2,2,2,2,2,2,2,2,1,1,2,2,2,2,1,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1].$$

We get

$$z_1 + ax_1^5x_2^3 + a^2x_1^4x_2^3 + a^4x_1^3x_2^3 + a^4x_1^2x_2^3 + a^2x_1x_2^3$$
$$+ax_2^3 + a^4x_1^6x_2^2 + a^4x_1^5x_2^2 + a^2x_1^4x_2^2 + ax_1^3x_2^2 +$$
$$a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^2x_1^3x_2 + a^4x_1x_2 + a^4x_1^4 + ax_1 + a$$

Configuration 4

Number of points	Third coordinate
7	a^5
6	a^6
5	1
4	a
3	a^2
2	a^3
1	a^4

The points are:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a, a^4)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$
$$[(a^3, a^5, a^5, a^2)],$$
$$[(1, a^3, a^6, a^2)],$$
$$[(a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^5, a^3)],$$
$$[(a^4, a, a^6, a^3)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^5, a^4)],$$
$$[(a^3, 1, a^6, a^4)],$$
$$[(a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

and their configuration is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^6	a^4,a^3,a^5	a^5,a^4,1	a^6,a^3,a^3	1,a^6,a^5
a,a^6,1	a^2,a^4,a^5	a^3,a^5,a^5	a^4,a,a^6	a^5,1,a^2	a^6,a^2,1	1,a^3,a^6
a,a,a^2	a^2,a^2,a	a^3,a,1	a^4,a^4,a	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

The configuration list is

$$[2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1],$$

corresponding to the polynomial

$$z_1 + a^2x_1^5x_2^3 + a^4x_1^4x_2^3 + x_1^3x_2^3 + ax_1^2x_2^3 + x_1x_2^3 + x_2^3 + x_1^6x_2^2 + ax_1^5x_2^2 + x_1^4x_2^2 + x_1^3x_2^2 + a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^4x_1^3x_2 + ax_1x_2 + ax_1^4 + ax_1 + a^2$$

Configuration 5

Number of points	Third coordinate
7	a^6
6	1
5	a
4	a^2
3	a^3
2	a^4
1	a^5

The points are:

$$[(a, a^3, a, 0)],$$
$$[(a^2, a^6, a^2, 0)],$$
$$[(a^3, a^2, a^3, 0)],$$
$$[(a^4, a^5, a^4, 0)],$$
$$[(a^5, a, a^5, 0)],$$
$$[(a^6, a^4, a^6, 0)],$$
$$[(1, 1, 1, 0)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a, a^4)],$$
$$[(a^6, 1, a, a^5)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$
$$[(a^3, a^5, a^2, a^5)],$$
$$[(1, a^3, a^6, a^2)],$$
$$[(a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^6, a^3)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^6, a^4)],$$
$$[(a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^6, a^5)],$$
$$[(a^4, a^3, 1, a^5)],$$
$$[(a^2, a^5, a^6, 1)].$$

and the tower structure is

a,a^2,a^6	a^2,a^5,a^6	$a^3,1,a^6$	$a^4,a^3,1$	$a^5,a^4,1$	a^6,a^3,a^3	$1,a^6,a^4$
$a,a^6,1$	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^6	$a^5,1,a^2$	$a^6,a^2,1$	$1,a^3,a^6$
a,a,a^2	a^2,a^2,a	$a^3,a,1$	a^4,a^4,a	a^5,a^6,a^6	$a^6,1,a$	$1,a^5,a$
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	$1,1,1$

Therefore, the configuration list is

$$[2,2,2,2,2,2,2,1,1,1,1,2,2,1,1,1,2,2,1,1,2,2,1,2,2,2,2,1]$$

and the polynomial we get is

$$\begin{aligned} &z_1 + a^3x_1^5x_2^3 + a^6x_1^4x_2^3 + a^3x_1^3x_2^3 + a^5x_1^2x_2^3 + a^5x_1x_2^3 + \\ &a^6x_2^3 + a^3x_1^6x_2^2 + a^5x_1^5x_2^2 + a^5x_1^4x_2^2 + a^6x_1^3x_2^2 + \\ &a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^6x_1^3x_2 + a^5x_1x_2 + a^5x_1^4 + ax_1 + a^3. \end{aligned} \tag{B.1}$$

Configuration 6

Number of points	Third coordinate
7	1
6	a
5	a^2
4	a^3
3	a^4
2	a^5
1	a^6

The points are:

$$[(a, a^3, a, 0)], \tag{B.2}$$
$$[(a^2, a^6, a^2, 0)], \tag{B.3}$$
$$[(a^3, a^2, a^3, 0)], \tag{B.4}$$
$$[(a^4, a^5, a^4, 0)], \tag{B.5}$$
$$[(a^5, a, a^5, 0)], \tag{B.6}$$
$$[(a^6, a^4, a^6, 0)], \tag{B.7}$$
$$[(1, 1, 1, 0)], \tag{B.8}$$
$$[(a^4, a^4, a, a^2)], \tag{B.9}$$
$$[(1, a^5, a, a^3)], \tag{B.10}$$
$$[(a^2, a^2, a, a^4)], \tag{B.11}$$
$$[(a^6, 1, a, a^5)], \tag{B.12}$$
$$[(a^5, a^6, a, a^6)], \tag{B.13}$$
$$[(a^3, a, 1, a)], \tag{B.14}$$
$$[(a^5, 1, a^2, a^3)], \tag{B.15}$$
$$[(a, a, a^2, a^4)], \tag{B.16}$$
$$[(a^3, a^5, a^2, a^5)], \tag{B.17}$$
$$[(1, a^3, a^2, a^6)], \tag{B.18}$$
$$[(a^6, a^2, a^2, 1)], \tag{B.19}$$
$$[(a^6, a^3, a^3, a^4)], \tag{B.20}$$
$$[(a^2, a^4, a^3, a^5)], \tag{B.21}$$
$$[(a^4, a, a^3, a^6)], \tag{B.22}$$
$$[(a, a^6, 1, a^3)], \tag{B.23}$$
$$[(1, a^6, a^4, a^5)], \tag{B.24}$$
$$[(a^3, 1, a^4, a^6)], \tag{B.25}$$
$$[(a^5, a^4, 1, a^4)], \tag{B.26}$$
$$[(a, a^2, a^5, a^6)], \tag{B.27}$$
$$[(a^4, a^3, 1, a^5)], \tag{B.28}$$
$$[(a^2, a^5, 1, a^6)]. \tag{B.29}$$

and the tower structure is

a,a^2,a^5	$a^2,a^5,1$	$a^3,1,a^4$	$a^4,a^3,1$	$a^5,a^4,1$	a^6,a^3,a^3	$1,a^6,a^4$
$a,a^6,1$	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	$a^5,1,a^2$	a^6,a^2,a^2	$1,a^3,a^2$
a,a,a^2	a^2,a^2,a	$a^3,a,1$	a^4,a^4,a	a^5,a^6,a	$a^6,1,a$	$1,a^5,a$
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	$1,1,1$

corresponding to the configuration list

$$[2,2,2,2,2,2,2,1,1,1,1,1,2,1,1,1,1,2,1,1,1,2,1,1,2,1,2,2].$$

The polynomial we get is

$$\begin{aligned} &z_1 + a^4x_1^5x_2^3 + ax_1^4x_2^3 + a^6x_1^3x_2^3 + a^2x_1^2x_2^3 + a^3x_1x_2^3 + \\ &a^5x_2^3 + a^6x_1^6x_2^2 + a^2x_1^5x_2^2 + a^3x_1^4x_2^2 + a^5x_1^3x_2^2 + \\ &a^6x_1^2x_2^2 + a^4x_1^5x_2 + ax_1^3x_2 + a^2x_1x_2 + a^2x_1^4 + ax_1 + a^4 \end{aligned} \tag{B.30}$$

Configuration 7

Number of points	Third coordinate
7	a
6	a^2
5	a^3
4	a^4
3	a^5
2	a^6
1	1

The points are:

$$[(a, a^3, a, 0)], \tag{B.31}$$
$$[(a^2, a^6, a^2, 0)], \tag{B.32}$$
$$[(a^3, a^2, a^3, 0)], \tag{B.33}$$
$$[(a^4, a^5, a^4, 0)], \tag{B.34}$$
$$[(a^5, a, a^5, 0)], \tag{B.35}$$
$$[(a^6, a^4, a^6, 0)], \tag{B.36}$$
$$[(1, 1, 1, 0)], \tag{B.37}$$
$$[(a^4, a^4, a, a^2)], \tag{B.38}$$
$$[(1, a^5, a, a^3)], \tag{B.39}$$
$$[(a^2, a^2, a, a^4)], \tag{B.40}$$
$$[(a^6, 1, a, a^5)], \tag{B.41}$$
$$[(a^5, a^6, a, a^6)], \tag{B.42}$$
$$[(a^3, a, a, 1)], \tag{B.43}$$
$$[(a^5, 1, a^2, a^3)], \tag{B.44}$$
$$[(a, a, a^2, a^4)], \tag{B.45}$$
$$[(a^3, a^5, a^2, a^5)], \tag{B.46}$$
$$[(1, a^3, a^2, a^6)], \tag{B.47}$$
$$[(a^6, a^2, a^2, 1)], \tag{B.48}$$
$$[(a^6, a^3, a^3, a^4)], \tag{B.49}$$
$$[(a^2, a^4, a^3, a^5)], \tag{B.50}$$
$$[(a^4, a, a^3, a^6)], \tag{B.51}$$
$$[(a, a^6, a^3, 1)], \tag{B.52}$$
$$[(1, a^6, a^4, a^5)], \tag{B.53}$$
$$[(a^3, 1, a^4, a^6)], \tag{B.54}$$
$$[(a^5, a^4, a^4, 1)], \tag{B.55}$$
$$[(a, a^2, a^5, a^6)], \tag{B.56}$$
$$[(a^4, a^3, a^5, 1)], \tag{B.57}$$
$$[(a^2, a^5, a^6, 1)]. \tag{B.58}$$

and their configuration list is

$$[2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1].$$

The tower structure is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^4	a^6,a^3,a^3	1,a^6,a^4	
a,a^6,a^3	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,a^2	1,a^3,a^2	
a,a,a^2	a^2,a^2,a	a^3,a,a	a^4,a^4,a	a^5,a^6,a	a^6,1,a	1,a^5,a	
a,a^3,a	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1	

and the corresponding polynomial is

$$\begin{aligned} &z_1 + a^5x_1^5x_2^3 + a^3x_1^4x_2^3 + a^2x_1^3x_2^3 + a^6x_1^2x_2^3 + \\ &ax_1x_2^3 + a^4x_2^3 + a^2x_1^6x_2^2 + a^6x_1^5x_2^2 + ax_1^4x_2^2 + a^4x_1^3x_2^2 \\ &+a^6x_1^2x_2^2 + a^4x_1^5x_2 + a^3x_1^3x_2 + a^6x_1x_2 + a^6x_1^4 + ax_1 + a^5 \end{aligned} \tag{B.59}$$

Here we give all the data for configuration 7.2 (8.3). It shows how the general error locator polynomial can change remarkably, even if we change only one point (the one marked in red).

Configuration 7.2

Number of points	Third coordinate
7	a^2
6	a^3
5	a^4
4	a^5
3	a^6
2	1
1	0

we get

$$[(a, a^3, 0, a)], \tag{B.60}$$
$$[(a^2, a^6, a^2, 0)], \tag{B.61}$$
$$[(a^3, a^2, a^3, 0)], \tag{B.62}$$
$$[(a^4, a^5, a^4, 0)], \tag{B.63}$$
$$[(a^5, a, a^5, 0)], \tag{B.64}$$
$$[(a^6, a^4, a^6, 0)], \tag{B.65}$$
$$[(1, 1, 1, 0)], \tag{B.66}$$
$$[(a^4, a^4, a^2, a)], \tag{B.67}$$
$$[(1, a^5, a^3, a)], \tag{B.68}$$
$$[(a^2, a^2, a^4, a)], \tag{B.69}$$
$$[(a^6, 1, a^5, a)], \tag{B.70}$$
$$[(a^5, a^6, a^6, a)], \tag{B.71}$$
$$[(a^3, a, 1, a)], \tag{B.72}$$
$$[(a^5, 1, a^2, a^3)], \tag{B.73}$$
$$[(a, a, a^2, a^4)], \tag{B.74}$$
$$[(a^3, a^5, a^2, a^5)], \tag{B.75}$$
$$[(1, a^3, a^2, a^6)], \tag{B.76}$$
$$[(a^6, a^2, a^2, 1)], \tag{B.77}$$
$$[(a^6, a^3, a^3, a^4)], \tag{B.78}$$
$$[(a^2, a^4, a^3, a^5)], \tag{B.79}$$
$$[(a^4, a, a^3, a^6)], \tag{B.80}$$
$$[(a, a^6, a^3, 1)], \tag{B.81}$$
$$[(1, a^6, a^4, a^5)], \tag{B.82}$$
$$[(a^3, 1, a^4, a^6)], \tag{B.83}$$
$$[(a^5, a^4, a^4, 1)], \tag{B.84}$$
$$[(a, a^2, a^5, a^6)], \tag{B.85}$$
$$[(a^4, a^3, a^5, 1)], \tag{B.86}$$
$$[(a^2, a^5, a^6, 1)]. \tag{B.87}$$

The tower structure is

a,a^2,a^5	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,a^4	a^6,a^3,a^3	1,a^6,a^4
a,a^6,a^3	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,a^2	1,a^3,a^2
a,a,a^2	a^2,a^2,a^4	a^3,a,1	a^4,a^4,a^2	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a^3
$\mathbf{a}$,$\mathbf{a^3}$,$\mathbf{0}$	a^2,a^6,a^2	a^3,a^2,a^3	a^4,a^5,a^4	a^5,a,a^5	a^6,a^4,a^6	1,1,1

and the configuration list is

$$[1,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1].$$

The polynomial we get, namely

$$\begin{gathered} z_1 + x_1^6 x_2^3 + a^5 x_1^5 x_2^3 + a^3 x_1^4 x_2^3 + a^2 x_1^3 x_2^3 + \\ a^6 x_1^2 x_2^3 + a x_1 x_2^3 + a^4 x_2^3 + a^2 x_1^6 x_2^2 + a^6 x_1^5 x_2^2 + \\ a x_1^4 x_2^2 + a^4 x_1^3 x_2^2 + a^2 x_1^2 x_2^2 + a x_1 x_2^2 + a^2 x_2^2 + \\ a^4 x_1^5 x_2 + a^5 x_1^3 x_2 + a^3 x_1 x_2 + a^2 x_1^6 + a^3 x_1^5 + a^6 x_1^4 + \\ a^5 x_1^3 + a^6 x_1^2 + a^3 x_1 + a^5 \end{gathered} \tag{B.88}$$

is made up of 25 terms.

B.1.2 Seven matrices, seven sets of formulas.

We list here the seven coefficient matrices for the cyclical configurations in $\mathbb{F}_8$ (8.3).

Configuration 1**:**

$$A^{[1]} = \begin{pmatrix} 0 & 0 & a^5 & 0 \\ 0 & a^4 & a^3 & a^6 \\ a^3 & 0 & a^6 & a^5 \\ 0 & a^5 & a^3 & a^5 \\ 0 & 0 & a^6 & a^3 \\ a & a^3 & 0 & a^6 \\ a^6 & 0 & 0 & a^3 \end{pmatrix}$$

Configuration 2**:**

$$A^{[2]} = \begin{pmatrix} 0 & 0 & a & 0 \\ 0 & a^4 & 1 & 1 \\ 1 & 0 & a^4 & 1 \\ 0 & 1 & a^2 & a \\ 0 & 0 & a^6 & 1 \\ a & 1 & 0 & a^4 \\ 1 & 0 & 0 & a^2 \end{pmatrix}$$

Configuration 3:

$$A^{[3]} = \begin{pmatrix} 0 & 0 & a^4 & 0 \\ 0 & a^4 & a^4 & a \\ a^4 & 0 & a^2 & a^2 \\ 0 & a^2 & a & a^4 \\ 0 & 0 & a^6 & a^4 \\ a & a^4 & 0 & a^2 \\ a & 0 & 0 & a \end{pmatrix}$$

Configuration 4:

$$A^{[4]} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & a^4 & a & a^2 \\ a & 0 & 1 & a^4 \\ 0 & a^4 & 1 & 1 \\ 0 & 0 & a^6 & a \\ a & a & 0 & 1 \\ a^2 & 0 & 0 & 1 \end{pmatrix}$$

Configuration 5:

$$A^{[5]} = \begin{pmatrix} 0 & 0 & a^3 & 0 \\ 0 & a^4 & a^3 & a^3 \\ a^5 & 0 & a^5 & a^6 \\ 0 & a^6 & a^6 & a^3 \\ 0 & 0 & a^6 & a^5 \\ a & a^5 & 0 & a^5 \\ a^3 & 0 & 0 & a^6 \end{pmatrix}$$

Configuration 6:

$$A^{[6]} = \begin{pmatrix} 0 & 0 & a^6 & 0 \\ 0 & a^4 & a^2 & a^4 \\ a^2 & 0 & a^3 & a \\ 0 & a & a^5 & a^6 \\ 0 & 0 & a^6 & a^2 \\ a & a^2 & 0 & a^3 \\ a^4 & 0 & 0 & a^5 \end{pmatrix}$$

Configuration 7:

$$\begin{pmatrix} 0 & 0 & a^2 & 0 \\ 0 & a^4 & a^6 & a^5 \\ a^6 & 0 & a & a^3 \\ 0 & a^3 & a^4 & a^2 \\ 0 & 0 & a^6 & a^6 \\ a & a^6 & 0 & a \\ a^5 & 0 & 0 & a^4 \end{pmatrix}$$

The "general" matrix of the coefficients is

$$A^{[\text{gen}]} = \begin{pmatrix} 0 & 0 & C & 0 \\ 0 & a^4 & D & A \\ D & 0 & E & B \\ 0 & B & F & C \\ 0 & 0 & a^6 & D \\ a & D & 0 & E \\ A & 0 & 0 & F \end{pmatrix}; \ A, B, C, D, E, F \in \mathbb{F}_8.$$

As explained in chapter 8, choosing differently the value M, we have different sets of formulas, summarized in the table below.

A-F	$M^?$	1	2	3	4	5	6	7
A	M	5	6	7	1	2	3	4
B	M^2	3	5	7	2	4	6	1
C	M^3	2	5	1	4	7	3	6
D	M^4	6	3	7	4	1	5	2
E	M^5	1	6	4	3	7	5	3
F	M^6	4	3	2	1	7	6	5

Table B.1: Configurations in $\mathbb{F}_8$.

As explained in 8.3, the formulas in the table above are linked with the cycles in $\mathbb{F}_8$. We list here the couples of cycles connected to each set of formulas.

1: $(\alpha, \beta), (\beta, \alpha)$;

2: $(\alpha, \beta), (\beta, \beta)$;

3: $(\alpha, \gamma), (\beta, \alpha)$;

4: $(\alpha, \alpha), (\beta, \alpha)$;

5: $(\alpha, \alpha), (\beta, \gamma)$;

6: $(\alpha, \beta), (\beta, \beta)$;

7: $(\alpha, \alpha), (\beta, \beta)$.

B.2 Cyclical configurations in $\mathbb{F}_{16}$.

B.2.1 The cyclical configurations.

Here we list the 15 cyclical configurations in $\mathbb{F}_{16}$ (8.4).

Configuration 1:

Number of points	Third coordinate
15	a
14	a^2
13	a^3
12	a^4
11	a^5
10	a^6
9	a^7
8	a^8
7	a^9
6	a^{10}
5	a^{11}
4	a^{12}
3	a^{13}
2	a^{14}
1	1

The configuration list is

$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$

and the associated locator polinomial is made up of 85 terms

$$z_1 + a^{14}x_1^{14}x_2^7 + ax_1^{13}x_2^7 + a^7x_1^{12}x_2^7 + a^{11}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{12}x_1^9x_2^7 + a^9x_1^8x_2^7 + a^4x_1^7x_2^7 +$$
$$a^3x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^8x_1^4x_2^7 + a^{13}x_1^3x_2^7 + a^6x_1^2x_2^7 + a^2x_1x_2^7 + a^5x_2^7 + a^{11}x_1^{14}x_2^6 +$$
$$a^5x_1^{13}x_2^6 + a^4x_1^{10}x_2^6 + a^3x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^8x_1^7x_2^6 + a^{13}x_1^6x_2^6 + a^6x_1^5x_2^6 + a^2x_1^4x_2^6 +$$
$$a^5x_1^3x_2^6 + a^{14}x_1^2x_2^6 + ax_1x_2^6 + a^7x_2^6 + a^4x_1^{13}x_2^5 + a^{10}x_1^{11}x_2^5 + a^8x_1^{10}x_2^5 + a^{13}x_1^9x_2^5 + a^6x_1^8x_2^5 +$$
$$a^2x_1^7x_2^5 + a^5x_1^6x_2^5 + a^{14}x_1^5x_2^5 + ax_1^4x_2^5 + a^7x_1^3x_2^5 + a^{11}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{12}x_2^5 + a^{13}x_1^{12}x_2^4 +$$
$$a^6x_1^{11}x_2^4 + a^2x_1^{10}x_2^4 + a^5x_1^9x_2^4 + a^{14}x_1^8x_2^4 + ax_1^7x_2^4 + a^7x_1^6x_2^4 + a^{11}x_1^5x_2^4 + a^9x_1^4x_2^4 +$$
$$a^9x_1^2x_2^4 + a^3x_2^4 + a^6x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^3x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^3x_1^6x_2^2 + a^{10}x_1^5x_2^2 +$$
$$a^8x_1^4x_2^2 + a^6x_1^2x_2^2 + a^2x_1x_2^2 + a^5x_2^2 + a^{11}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{12}x_1^{12}x_2 + a^9x_1^{11}x_2 +$$
$$a^4x_1^{10}x_2 + a^{10}x_1^8x_2 + a^{13}x_1^6x_2 + a^6x_1^5x_2 + a^2x_1^4x_2 + a^{14}x_1^2x_2 + a^7x_2 + a^9x_1^{14} + a^3x_1^{12} +$$
$$a^{10}x_1^{11} + a^{13}x_1^9 + a^6x_1^8 + a^5x_1^6 + a^{14}x_1^5 + a^7x_1^3 + a^{11}x_1^2 + a^{10}x_1 + a^{12}$$

Configuration 2**:**

Number of points	Third coordinate
15	1
14	a
13	a^2
12	a^3
11	a^4
10	a^5
9	a^6
8	a^7
7	a^8
6	a^9
5	a^{10}
4	a^{11}
3	a^{12}
2	a^{13}
1	a^{14}

The configuration's list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,$$
$$1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,$$
$$1,1,2,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1,1,2,1,1,1,1,1,1,2,1,1,1,1,1,2,1,1,1,$$
$$1,2,1,1,1,2,1,1,2,1,2,2.$$

The corresponding locator polynomial is

$$z_1 + a^3x_1^{14}x_2^7 + a^4x_1^{13}x_2^7 + a^9x_1^{12}x_2^7 + a^{12}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{11}x_1^9x_2^7 + a^7x_1^8x_2^7 + ax_1^7x_2^7+$$
$$a^{14}x_1^6x_2^7 + a^5x_1^5x_2^7 + a^2x_1^4x_2^7 + a^6x_1^3x_2^7 + a^{13}x_1^2x_2^7 + a^8x_1x_2^7 + a^{10}x_2^7+$$
$$a^{12}x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + ax_1^{10}x_2^6 + a^{14}x_1^9x_2^6 + a^5x_1^8x_2^6 + a^2x_1^7x_2^6 + a^6x_1^6x_2^6+$$
$$a^{13}x_1^5x_2^6 + a^8x_1^4x_2^6 + a^{10}x_1^3x_2^6 + a^3x_1^2x_2^6 + a^4x_1x_2^6 + a^9x_2^6 + ax_1^{13}x_2^5 + a^5x_1^{11}x_2^5+$$
$$a^2x_1^{10}x_2^5 + a^6x_1^9x_2^5 + a^{13}x_1^8x_2^5 + a^8x_1^7x_2^5 + a^{10}x_1^6x_2^5 + a^3x_1^5x_2^5 + a^4x_1^4x_2^5+$$
$$a^9x_1^3x_2^5 + a^{12}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{11}x_2^5 + a^6x_1^{12}x_2^4 + a^{13}x_1^{11}x_2^4 + a^8x_1^{10}x_2^4 + a^{10}x_1^9x_2^4+$$
$$a^3x_1^8x_2^4 + a^4x_1^7x_2^4 + a^9x_1^6x_2^4 + a^{12}x_1^5x_2^4 + a^9x_1^4x_2^4 + a^7x_1^2x_2^4+$$
$$a^{14}x_2^4 + a^{13}x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^{14}x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^{14}x_1^6x_2^2 + a^5x_1^5x_2^2+$$
$$a^2x_1^4x_2^2 + a^{13}x_1^2x_2^2 + a^8x_1x_2^2 + a^{10}x_2^2 + a^{12}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{11}x_1^{12}x_2+$$
$$a^7x_1^{11}x_2 + ax_1^{10}x_2 + a^5x_1^8x_2 + a^6x_1^6x_2 + a^{13}x_1^5x_2 + a^8x_1^4x_2 + a^3x_1^2x_2 + a^9x_2+$$
$$a^7x_1^{14} + a^{14}x_1^{12} + a^5x_1^{11} + a^6x_1^9 + a^{13}x_1^8 + a^{10}x_1^6 + a^3x_1^5 + a^9x_1^3 + a^{12}x_1^2 + a^{10}x_1 + a^{11}$$

Configuration 3**:**

Number of points	Third coordinate
15	a^{14}
14	1
13	a
12	a^2
11	a^3
10	a^4
9	a^5
8	a^6
7	a^7
6	a^8
5	a^9
4	a^{10}
3	a^{11}
2	a^{12}
1	a^{13}

The configuration's list is

$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1,$$
$$1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 1$$

and the locator polynomial is

$$z_1 + a^7x_1^{14}x_2^7 + a^7x_1^{13}x_2^7 + a^{11}x_1^{12}x_2^7 + a^{13}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{10}x_1^9x_2^7 + a^5x_1^8x_2^7 + a^{13}x_1^7x_2^7 + a^{10}x_1^6x_2^7 + x_1^5x_2^7 + a^{11}x_1^4x_2^7 + a^{14}x_1^3x_2^7 + a^5x_1^2x_2^7 + a^{14}x_1x_2^7 + x_2^7 + a^{13}x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^{13}x_1^{10}x_2^6 + a^{10}x_1^9x_2^6 + x_1^8x_2^6 + a^{11}x_1^7x_2^6 + a^{14}x_1^6x_2^6 + a^5x_1^5x_2^6 + a^{14}x_1^4x_2^6 + x_1^3x_2^6 + a^7x_1^2x_2^6 + a^7x_1x_2^6 + a^{11}x_2^6 + a^{13}x_1^{13}x_2^5 + x_1^{11}x_2^5 + a^{11}x_1^{10}x_2^5 + a^{14}x_1^9x_2^5 + a^5x_1^8x_2^5 + a^{14}x_1^7x_2^5 + x_1^6x_2^5 + a^7x_1^5x_2^5 + a^7x_1^4x_2^5 + a^{11}x_1^3x_2^5 + a^{13}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{10}x_2^5 + a^{14}x_1^{12}x_2^4 + a^5x_1^{11}x_2^4 + a^{14}x_1^{10}x_2^4 + x_1^9x_2^4 + a^7x_1^8x_2^4 + a^7x_1^7x_2^4 + a^{11}x_1^6x_2^4 + a^{13}x_1^5x_2^4 + a^9x_1^4x_2^4 + a^5x_1^2x_2^4 + a^{10}x_2^4 + a^5x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^{10}x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^{10}x_1^6x_2^2 + x_1^5x_2^2 + a^{11}x_1^4x_2^2 + a^5x_1^2x_2^2 + a^{14}x_1x_2^2 + x_2^2 + a^{13}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{10}x_1^{12}x_2 + a^5x_1^{11}x_2 + a^{13}x_1^{10}x_2 + x_1^8x_2 + a^{14}x_1^6x_2 + a^5x_1^5x_2 + a^{14}x_1^4x_2 + a^7x_1^2x_2 + a^{11}x_2 + a^5x_1^{14} + a^{10}x_1^{12} + x_1^{11} + a^{14}x_1^9 + a^5x_1^8 + x_1^6 + a^7x_1^5 + a^{11}x_1^3 + a^{13}x_1^2 + a^{10}x_1 + a^{10}$$

Configuration 4**:**

Number of points	Third coordinate
15	a^{13}
14	a^{14}
13	1
12	a
11	a^2
10	a^3
9	a^4
8	a^5
7	a^6
6	a^7
5	a^8
4	a^9
3	a^{10}
2	a^{11}
1	a^{12}

The configuration list is

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, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1

The locator polynomial is

$$z_1 + a^{11}x_1^{14}x_2^7 + a^{10}x_1^{13}x_2^7 + a^{13}x_1^{12}x_2^7 + a^{14}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^9x_1^9x_2^7 + a^3x_1^8x_2^7 + a^{10}x_1^7x_2^7 + a^6x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^5x_1^4x_2^7 + a^7x_1^3x_2^7 + a^{12}x_1^2x_2^7 + a^5x_1x_2^7 + a^5x_2^7 + a^{14}x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^{10}x_1^{10}x_2^6 + a^6x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^5x_1^7x_2^6 + a^7x_1^6x_2^6 + a^{12}x_1^5x_2^6 + a^5x_1^4x_2^6 + a^5x_1^3x_2^6 + a^{11}x_1^2x_2^6 + a^{10}x_1x_2^6 + a^{13}x_2^6 + a^{10}x_1^{13}x_2^5 + a^{10}x_1^{11}x_2^5 + a^5x_1^{10}x_2^5 + a^7x_1^9x_2^5 + a^{12}x_1^8x_2^5 + a^5x_1^7x_2^5 + a^5x_1^6x_2^5 + a^{11}x_1^5x_2^5 + a^{10}x_1^4x_2^5 + a^{13}x_1^3x_2^5 + a^{14}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^9x_2^5 + a^7x_1^{12}x_2^4 + a^{12}x_1^{11}x_2^4 + a^5x_1^{10}x_2^4 + a^5x_1^9x_2^4 + a^{11}x_1^8x_2^4 + a^{10}x_1^7x_2^4 + a^{13}x_1^6x_2^4 + a^{14}x_1^5x_2^4 + a^9x_1^4x_2^4 + a^3x_1^2x_2^4 + a^6x_2^4 + a^{12}x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^6x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^6x_1^6x_2^2 + a^{10}x_1^5x_2^2 + a^5x_1^4x_2^2 + a^{12}x_1^2x_2^2 + a^5x_1x_2^2 + a^5x_2^2 + a^{14}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^9x_1^{12}x_2 + a^3x_1^{11}x_2 + a^{10}x_1^{10}x_2 + a^{10}x_1^8x_2 + a^7x_1^6x_2 + a^{12}x_1^5x_2 + a^5x_1^4x_2 + a^{11}x_1^2x_2 + a^{13}x_2 + a^3x_1^{14} + a^6x_1^{12} + a^{10}x_1^{11} + a^7x_1^9 + a^{12}x_1^8 + a^5x_1^6 + a^{11}x_1^5 + a^{13}x_1^3 + a^{14}x_1^2 + a^{10}x_1 + a^9$$

Configuration 5**:**

Number of points	Third coordinate
15	a^{12}
14	a^{13}
13	a^{14}
12	1
11	a
10	a^2
9	a^3
8	a^4
7	a^5
6	a^6
5	a^7
4	a^8
3	a^9
2	a^{10}
1	a^{11}

The configuration list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,2,2,2,2,1,1,1,1,1,1,1,1,1,2,2,$$
$$2,2,1,1,1,1,1,1,1,1,2,2,2,2,1,1,1,1,1,1,1,2,2,2,2,1,1,1,1,1,2,2,2,2,1,1,1,1,1,$$
$$2,2,2,2,1,1,1,1,1,2,2,2,2,1,1,1,2,2,2,2,1,1,2,2,2,2,1,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1$$

and the locator polynomial is

$$z_1 + x_1^{14}x_2^7 + a^{13}x_1^{13}x_2^7 + x_1^{12}x_2^7 + x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^8x_1^9x_2^7 + ax_1^8x_2^7 + a^7x_1^7x_2^7 + a^2x_1^6x_2^7 + a^5x_1^5x_2^7 + a^{14}x_1^4x_2^7 + x_1^3x_2^7 + a^4x_1^2x_2^7 + a^{11}x_1x_2^7 + a^{10}x_2^7 + x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^7x_1^{10}x_2^6 + a^2x_1^9x_2^6 + a^5x_1^8x_2^6 + a^{14}x_1^7x_2^6 + x_1^6x_2^6 + a^4x_1^5x_2^6 + a^{11}x_1^4x_2^6 + a^{10}x_1^3x_2^6 + x_1^2x_2^6 + a^{13}x_1x_2^6 + x_2^6 + a^7x_1^{13}x_2^5 + a^5x_1^{11}x_2^5 + a^{14}x_1^{10}x_2^5 + x_1^9x_2^5 + a^4x_1^8x_2^5 + a^{11}x_1^7x_2^5 + a^{10}x_1^6x_2^5 + x_1^5x_2^5 + a^{13}x_1^4x_2^5 + x_1^3x_2^5 + x_1^2x_2^5 + a^{10}x_1x_2^5 + a^8x_2^5 + x_1^{12}x_2^4 + a^4x_1^{11}x_2^4 + a^{11}x_1^{10}x_2^4 + a^{10}x_1^9x_2^4 + x_1^8x_2^4 + a^{13}x_1^7x_2^4 + x_1^6x_2^4 + x_1^5x_2^4 + a^9x_1^4x_2^4 + ax_1^2x_2^4 + a^2x_2^4 + a^4x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^2x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^2x_1^6x_2^2 + a^5x_1^5x_2^2 + a^{14}x_1^4x_2^2 + a^4x_1^2x_2^2 + a^{11}x_1x_2^2 + a^{10}x_2^2 + x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^8x_1^{12}x_2 + ax_1^{11}x_2 + a^7x_1^{10}x_2 + a^5x_1^8x_2 + x_1^6x_2 + a^4x_1^5x_2 + a^{11}x_1^4x_2 + x_1^2x_2 + x_2 + ax_1^{14} + a^2x_1^{12} + a^5x_1^{11} + x_1^9 + a^4x_1^8 + a^{10}x_1^6 + x_1^5 + x_1^3 + x_1^2 + a^{10}x_1 + a^8$$

Configuration 6**:**

Number of points	Third coordinate
15	a^{11}
14	a^{12}
13	a^{13}
12	a^{14}
11	1
10	a
9	a^2
8	a^3
7	a^4
6	a^5
5	a^6
4	a^7
3	a^8
2	a^9
1	a^{10}

The configuration list is

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, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2
2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2,
2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

The locator polynomial is

$$z_1 + a^4x_1^{14}x_2^7 + ax_1^{13}x_2^7 + a^2x_1^{12}x_2^7 + ax_1^{11}x_2^7 + x_1^{10}x_2^7 + a^7x_1^9x_2^7 + a^{14}x_1^8x_2^7 + a^4x_1^7x_2^7 + a^{13}x_1^6x_2^7 + x_1^5x_2^7 + a^8x_1^4x_2^7 + a^8x_1^3x_2^7 + a^{11}x_1^2x_2^7 + a^2x_1x_2^7 + x_2^7 + ax_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^4x_1^{10}x_2^6 + a^{13}x_1^9x_2^6 + x_1^8x_2^6 + a^8x_1^7x_2^6 + a^8x_1^6x_2^6 + a^{11}x_1^5x_2^6 + a^2x_1^4x_2^6 + x_1^3x_2^6 + a^4x_1^2x_2^6 + ax_1x_2^6 + a^2x_2^6 + a^4x_1^{13}x_2^5 + x_1^{11}x_2^5 + a^8x_1^{10}x_2^5 + a^8x_1^9x_2^5 + a^{11}x_1^8x_2^5 + a^2x_1^7x_2^5 + x_1^6x_2^5 + a^4x_1^5x_2^5 + ax_1^4x_2^5 + a^2x_1^3x_2^5 + ax_1^2x_2^5 + a^{10}x_1x_2^5 + a^7x_2^5 + a^8x_1^{12}x_2^4 + a^{11}x_1^{11}x_2^4 + a^2x_1^{10}x_2^4 + x_1^9x_2^4 + a^4x_1^8x_2^4 + ax_1^7x_2^4 + a^2x_1^6x_2^4 + ax_1^5x_2^4 + a^9x_1^4x_2^4 + a^{14}x_1^2x_2^4 + a^{13}x_2^4 + a^{11}x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^{13}x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^{13}x_1^6x_2^2 + x_1^5x_2^2 + a^8x_1^4x_2^2 + a^{11}x_1^2x_2^2 + a^2x_1x_2^2 + x_2^2 + ax_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^7x_1^{12}x_2 + a^{14}x_1^{11}x_2 + a^4x_1^{10}x_2 + x_1^8x_2 + a^8x_1^6x_2 + a^{11}x_1^5x_2 + a^2x_1^4x_2 + a^4x_1^2x_2 + a^2x_2 + a^{14}x_1^{14} + a^{13}x_1^{12} + x_1^{11} + a^8x_1^9 + a^{11}x_1^8 + x_1^6 + a^4x_1^5 + a^2x_1^3 + ax_1^2 + a^{10}x_1 + a^7$$

Configuration 7:

Number of points	Third coordinate
15	a^{10}
14	a^{11}
13	a^{12}
12	a^{13}
11	a^{14}
10	1
9	a
8	a^2
7	a^3
6	a^4
5	a^5
4	a^6
3	a^7
2	a^8
1	a^9

The configuration list is

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2,
2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

and the locator polynomial is

$$z_1 + a^8x_1^{14}x_2^7 + a^4x_1^{13}x_2^7 + a^4x_1^{12}x_2^7 + a^2x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^6x_1^9x_2^7 + a^{12}x_1^8x_2^7 + ax_1^7x_2^7 +$$
$$a^9x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^2x_1^4x_2^7 + ax_1^3x_2^7 + a^3x_1^2x_2^7 + a^8x_1x_2^7 + a^5x_2^7 + a^2x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 +$$
$$ax_1^{10}x_2^6 + a^9x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^2x_1^7x_2^6 + ax_1^6x_2^6 + a^3x_1^5x_2^6 + a^8x_1^4x_2^6 + a^5x_1^3x_2^6 + a^8x_1^2x_2^6 +$$
$$a^4x_1x_2^6 + a^4x_2^6 + ax_1^{13}x_2^5 + a^{10}x_1^{11}x_2^5 + a^2x_1^{10}x_2^5 + ax_1^9x_2^5 + a^3x_1^8x_2^5 + a^8x_1^7x_2^5 + a^5x_1^6x_2^5 +$$
$$a^8x_1^5x_2^5 + a^4x_1^4x_2^5 + a^4x_1^3x_2^5 + a^2x_1^2x_2^5 + a^{10}x_1x_2^5 + a^6x_2^5 + ax_1^{12}x_2^4 + a^3x_1^{11}x_2^4 + a^8x_1^{10}x_2^4 +$$
$$a^5x_1^9x_2^4 + a^8x_1^8x_2^4 + a^4x_1^7x_2^4 + a^4x_1^6x_2^4 + a^2x_1^5x_2^4 + a^9x_1^4x_2^4 + a^{12}x_1^2x_2^4 + a^9x_2^4 + a^3x_1^{14}x_2^3 +$$
$$a^{10}x_1^7x_2^3 + a^9x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^9x_1^6x_2^2 + a^{10}x_1^5x_2^2 + a^2x_1^4x_2^2 + a^3x_1^2x_2^2 + a^8x_1x_2^2 + a^5x_2^2 +$$
$$a^2x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^6x_1^{12}x_2 + a^{12}x_1^{11}x_2 + ax_1^{10}x_2 + a^{10}x_1^8x_2 + ax_1^6x_2 + a^3x_1^5x_2$$
$$+a^8x_1^4x_2 + a^8x_1^2x_2 + a^4x_2 + a^{12}x_1^{14} + a^9x_1^{12} + a^{10}x_1^{11} + ax_1^9 + a^3x_1^8 +$$
$$a^5x_1^6 + a^8x_1^5 + a^4x_1^3 + a^2x_1^2 + a^{10}x_1 + a^6$$

Configuration 8**:**

Number of points	Third coordinate
15	a^9
14	a^{10}
13	a^{11}
12	a^{12}
11	a^{13}
10	a^{14}
9	1
8	a
7	a^2
6	a^3
5	a^4
4	a^5
3	a^6
2	a^7
1	a^8

The coefficients' list is

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,
2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2,
2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

and the locator is

$$z_1 + a^{12}x_1^{14}x_2^7 + a^7x_1^{13}x_2^7 + a^6x_1^{12}x_2^7 + a^3x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^5x_1^9x_2^7 + a^{10}x_1^8x_2^7 + a^{13}x_1^7x_2^7 + a^5x_1^6x_2^7 + a^5x_1^5x_2^7 + a^{11}x_1^4x_2^7 + a^9x_1^3x_2^7 + a^{10}x_1^2x_2^7 + a^{14}x_1x_2^7 + a^{10}x_2^7 + a^3x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^{13}x_1^{10}x_2^6 + a^5x_1^9x_2^6 + a^5x_1^8x_2^6 + a^{11}x_1^7x_2^6 + a^9x_1^6x_2^6 + a^{10}x_1^5x_2^6 + a^{14}x_1^4x_2^6 + a^{10}x_1^3x_2^6 + a^{12}x_1^2x_2^6 + a^7x_1x_2^6 + a^6x_2^6 + a^{13}x_1^{13}x_2^5 + a^5x_1^{11}x_2^5 + a^{11}x_1^{10}x_2^5 + a^9x_1^9x_2^5 + a^{10}x_1^8x_2^5 + a^{14}x_1^7x_2^5 + a^{10}x_1^6x_2^5 + a^{12}x_1^5x_2^5 + a^7x_1^4x_2^5 + a^6x_1^3x_2^5 + a^3x_1^2x_2^5 + a^{10}x_1x_2^5 + a^5x_2^5 + a^9x_1^{12}x_2^4 + a^{10}x_1^{11}x_2^4 + a^{14}x_1^{10}x_2^4 + a^{10}x_1^9x_2^4 + a^{12}x_1^8x_2^4 + a^7x_1^7x_2^4 + a^6x_1^6x_2^4 + a^3x_1^5x_2^4 + a^9x_1^4x_2^4 + a^{10}x_1^2x_2^4 + a^5x_2^4 + a^{10}x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^5x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^5x_1^6x_2^2 + a^5x_1^5x_2^2 + a^{11}x_1^4x_2^2 + a^{10}x_1^2x_2^2 + a^{14}x_1x_2^2 + a^{10}x_2^2 + a^3x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^5x_1^{12}x_2 + a^{10}x_1^{11}x_2 + a^{13}x_1^{10}x_2 + a^5x_1^8x_2 + a^9x_1^6x_2 + a^{10}x_1^5x_2 + a^{14}x_1^4x_2 + a^{12}x_1^2x_2 + a^6x_2 + a^{10}x_1^{14} + a^5x_1^{12} + a^5x_1^{11} + a^9x_1^9 + a^{10}x_1^8 + a^{10}x_1^6 + a^{12}x_1^5 + a^6x_1^3 + a^3x_1^2 + a^{10}x_1 + a^5$$

Configuration 9**:**

Number of points	Third coordinate
15	a^8
14	a^9
13	a^{10}
12	a^{11}
11	a^{12}
10	a^{13}
9	a^{14}
8	1
7	a
6	a^2
5	a^3
4	a^4
3	a^5
2	a^6
1	a^7

The configuration's list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,2,2,2,2,2,2,2,2,1,1,1,1,1,2,2,2,2,2,2,$$
$$2,2,1,1,1,1,2,2,2,2,2,2,2,2,2,1,1,1,2,2,2,2,2,2,2,2,2,1,1,2,2,2,2,2,2,2,2,1,2,2,2,2,$$
$$2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1$$

and the locator is

$$z_1 + ax_1^{14}x_2^7 + a^{10}x_1^{13}x_2^7 + a^8x_1^{12}x_2^7 + a^4x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^4x_1^9x_2^7 + a^8x_1^8x_2^7 + a^{10}x_1^7x_2^7 + ax_1^6x_2^7 + x_1^5x_2^7 + a^5x_1^4x_2^7 + a^2x_1^3x_2^7 + a^2x_1^2x_2^7 + a^5x_1x_2^7 + x_2^7 + a^4x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^{10}x_1^{10}x_2^6 + ax_1^9x_2^6 + x_1^8x_2^6 + a^5x_1^7x_2^6 + a^2x_1^6x_2^6 + a^2x_1^5x_2^6 + a^5x_1^4x_2^6 + x_1^3x_2^6 + ax_1^2x_2^6 + a^{10}x_1x_2^6 + a^8x_2^6 + a^{10}x_1^{13}x_2^5 + x_1^{11}x_2^5 + a^5x_1^{10}x_2^5 + a^2x_1^9x_2^5 + a^2x_1^8x_2^5 + a^5x_1^7x_2^5 + x_1^6x_2^5 + ax_1^5x_2^5 + a^{10}x_1^4x_2^5 + a^8x_1^3x_2^5 + a^4x_1^2x_2^5 + a^{10}x_1x_2^5 + a^4x_2^5 + a^2x_1^{12}x_2^4 + a^2x_1^{11}x_2^4 + a^5x_1^{10}x_2^4 + x_1^9x_2^4 + ax_1^8x_2^4 + a^{10}x_1^7x_2^4 + a^8x_1^6x_2^4 + a^4x_1^5x_2^4 + a^9x_1^4x_2^4 + a^8x_1^2x_2^4 + ax_2^4 + a^2x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + ax_1^3x_2^3 + a^2x_1^{10}x_2^2 + ax_1^6x_2^2 + x_1^5x_2^2 + a^5x_1^4x_2^2 + a^2x_1^2x_2^2 + a^5x_1x_2^2 + x_2^2 + a^4x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^4x_1^{12}x_2 + a^8x_1^{11}x_2 + a^{10}x_1^{10}x_2 + x_1^8x_2 + a^2x_1^6x_2 + a^2x_1^5x_2 + a^5x_1^4x_2 + ax_1^2x_2 + a^8x_2 + a^8x_1^{14} + ax_1^{12} + x_1^{11} + a^2x_1^9 + a^2x_1^8 + x_1^6 + ax_1^5 + a^8x_1^3 + a^4x_1^2 + a^{10}x_1 + a^4$$

Configuration 10**:**

Number of points	Third coordinate
15	a^7
14	a^8
13	a^9
12	a^{10}
11	a^{11}
10	a^{12}
9	a^{13}
8	a^{14}
7	1
6	a
5	a^2
4	a^3
3	a^4
2	a^5
1	a^6

The configuration's list is

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

and the locator is

$$z_1 + a^5x_1^q14x_2^7 + a^{13}x_1^{13}x_2^7 + a^{10}x_1^{12}x_2^7 + a^5x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^3x_1^9x_2^7 + a^6x_1^8x_2^7 + a^7x_1^7x_2^7 + a^{12}x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^{14}x_1^4x_2^7 + a^{10}x_1^3x_2^7 + a^9x_1^2x_2^7 + a^{11}x_1x_2^7 + a^5x_2^7 + a^5x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^7x_1^{10}x_2^6 + a^{12}x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^{14}x_1^7x_2^6 + a^{10}x_1^6x_2^6 + a^9x_1^5x_2^6 + a^{11}x_1^4x_2^6 + a^5x_1^3x_2^6 + a^5x_1^2x_2^6 + a^{13}x_1x_2^6 + a^{10}x_2^6 + a^7x_1^{13}x_2^5 + a^{10}x_1^{11}x_2^5 + a^{14}x_1^{10}x_2^5 + a^{10}x_1^9x_2^5 + a^9x_1^8x_2^5 + a^{11}x_1^7x_2^5 + a^5x_1^6x_2^5 + a^5x_1^5x_2^5 + a^{13}x_1^4x_2^5 + a^{10}x_1^3x_2^5 + a^5x_1^2x_2^5 + a^{10}x_1x_2^5 + a^3x_2^5 + a^{10}x_1^{12}x_2^4 + a^9x_1^{11}x_2^4 + a^{11}x_1^{10}x_2^4 + a^5x_1^9x_2^4 + a^5x_1^8x_2^4 + a^{13}x_1^7x_2^4 + a^{10}x_1^6x_2^4 + a^5x_1^5x_2^4 + a^9x_1^4x_2^4 + a^6x_1^2x_2^4 + a^{12}x_2^4 + a^9x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^{12}x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^{12}x_1^6x_2^2 + a^{10}x_1^5x_2^2 + a^{14}x_1^4x_2^2 + a^9x_1^2x_2^2 + a^{11}x_1x_2^2 + a^5x_2^2 + a^5x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^3x_1^{12}x_2 + a^6x_1^{11}x_2 + a^7x_1^{10}x_2 + a^{10}x_1^8x_2 + a^{10}x_1^6x_2 + a^9x_1^5x_2 + a^{11}x_1^4x_2 + a^5x_1^2x_2 + a^{10}x_2 + a^6x_1^{14} + a^{12}x_1^{12} + a^{10}x_1^{11} + a^{10}x_1^9 + a^9x_1^8 + a^5x_1^6 + a^5x_1^5 + a^{10}x_1^3 + a^5x_1^2 + a^{10}x_1 + a^3$$

Configuration 11**:**

Number of points	Third coordinate
15	a^6
14	a^7
13	a^8
12	a^9
11	a^{10}
10	a^{11}
9	a^{12}
8	a^{13}
7	a^{14}
6	1
5	a
4	a^2
3	a^3
2	a^4
1	a^5

The configuration list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,1,1,1,2,2,2,2,2,2,2,$$
$$2,2,2,1,1,2,2,2,2,2,2,2,2,2,2,2,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,$$
$$1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1$$

and the locator is

$$z_1 + a^9x_1^{14}x_2^7 + ax_1^{13}x_2^7 + a^{12}x_1^{12}x_2^7 + a^6x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^2x_1^9x_2^7 + a^4x_1^8x_2^7 + a^4x_1^7x_2^7 + a^8x_1^6x_2^7 + a^5x_1^5x_2^7 + a^8x_1^4x_2^7 + a^3x_1^3x_2^7 + ax_1^2x_2^7 + a^2x_1x_2^7 + a^{10}x_2^7 + a^6x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^4x_1^{10}x_2^6 + a^8x_1^9x_2^6 + a^5x_1^8x_2^6 + a^8x_1^7x_2^6 + a^3x_1^6x_2^6 + ax_1^5x_2^6 + a^2x_1^4x_2^6 + a^{10}x_1^3x_2^6 + a^9x_1^2x_2^6 + ax_1x_2^6 + a^{12}x_2^6 + a^4x_1^{13}x_2^5 + a^5x_1^{11}x_2^5 + a^8x_1^{10}x_2^5 + a^3x_1^9x_2^5 + ax_1^8x_2^5 + a^2x_1^7x_2^5 + a^{10}x_1^6x_2^5 + a^9x_1^5x_2^5 + ax_1^4x_2^5 + a^{12}x_1^3x_2^5 + a^6x_1^2x_2^5 + a^{10}x_1x_2^5 + a^2x_2^5 + a^3x_1^{12}x_2^4 + ax_1^{11}x_2^4 + a^2x_1^{10}x_2^4 + a^{10}x_1^9x_2^4 + a^9x_1^8x_2^4 + ax_1^7x_2^4 + a^{12}x_1^6x_2^4 + a^6x_1^5x_2^4 + a^9x_1^4x_2^4 + a^4x_1^2x_2^4 + a^8x_2^4 + ax_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^8x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^8x_1^6x_2^2 + a^5x_1^5x_2^2 + a^8x_1^4x_2^2 + ax_1^2x_2^2 + a^2x_1x_2^2 + a^{10}x_2^2 + a^6x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^2x_1^{12}x_2 + a^4x_1^{11}x_2 + a^4x_1^{10}x_2 + a^5x_1^8x_2 + a^3x_1^6x_2 + ax_1^5x_2 + a^2x_1^4x_2 + a^9x_1^2x_2 + a^{12}x_2 + a^4x_1^{14} + a^8x_1^{12} + a^5x_1^{11} + a^3x_1^9 + ax_1^8 + a^{10}x_1^6 + a^9x_1^5 + a^{12}x_1^3 + a^6x_1^2 + a^{10}x_1 + a^2$$

Configuration 12**:**

Number of points	Third coordinate
15	a^5
14	a^6
13	a^7
12	a^8
11	a^9
10	a^{10}
9	a^{11}
8	a^{12}
7	a^{13}
6	a^{14}
5	1
4	a
3	a^2
2	a^3
1	a^4

The configuration list is

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 1, 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, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

and the locator is

$$z_1 + a^{13}x_1^{14}x_2^7 + a^4x_1^{13}x_2^7 + a^{14}x_1^{12}x_2^7 + a^7x_1^{11}x_2^7 + x_1^{10}x_2^7 + ax_1^9x_2^7 + a^2x_1^8x_2^7 +$$
$$ax_1^7x_2^7 + a^4x_1^6x_2^7 + x_1^5x_2^7 + a^2x_1^4x_2^7 + a^{11}x_1^3x_2^7 + a^8x_1^2x_2^7 + a^8x_1x_2^7 + x_2^7 +$$
$$a^7x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + ax_1^{10}x_2^6 + a^4x_1^9x_2^6 + x_1^8x_2^6 + a^2x_1^7x_2^6 + a^{11}x_1^6x_2^6 + a^8x_1^5x_2^6 +$$
$$a^8x_1^4x_2^6 + x_1^3x_2^6 + a^{13}x_1^2x_2^6 + a^4x_1x_2^6 + a^{14}x_2^6 + ax_1^{13}x_2^5 + x_1^{11}x_2^5 + a^2x_1^{10}x_2^5 + a^{11}x_1^9x_2^5 +$$
$$a^8x_1^8x_2^5 + a^8x_1^7x_2^5 + x_1^6x_2^5 + a^{13}x_1^5x_2^5 + a^4x_1^4x_2^5 + a^{14}x_1^3x_2^5 + a^7x_1^2x_2^5 + a^{10}x_1x_2^5 +$$
$$ax_2^5 + a^{11}x_1^{12}x_2^4 + a^8x_1^{11}x_2^4 + a^8x_1^{10}x_2^4 + x_1^9x_2^4 + a^{13}x_1^8x_2^4 + a^4x_1^7x_2^4 + a^{14}x_1^6x_2^4 +$$
$$a^7x_1^5x_2^4 + a^9x_1^4x_2^4 + a^2x_1^2x_2^4 + a^4x_2^4 + a^8x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^4x_1^3x_2^3 + a^2x_1^{10}x_2^2 +$$
$$a^4x_1^6x_2^2 + x_1^5x_2^2 + a^2x_1^4x_2^2 + a^8x_1^2x_2^2 + a^8x_1x_2^2 + x_2^2 + a^7x_1^{14}x_2 + a^{12}x_1^{13}x_2 + ax_1^{12}x_2 +$$
$$a^2x_1^{11}x_2 + ax_1^{10}x_2 + x_1^8x_2 + a^{11}x_1^6x_2 + a^8x_1^5x_2 + a^8x_1^4x_2 + a^{13}x_1^2x_2 + a^{14}x_2 +$$
$$a^2x_1^{14} + a^4x_1^{12} + x_1^{11} + a^{11}x_1^9 + a^8x_1^8 + x_1^6 + a^{13}x_1^5 + a^{14}x_1^3 + a^7x_1^2 + a^{10}x_1 + a$$

Configuration 13**:**

Number of points	Third coordinate
15	a^4
14	a^5
13	a^6
12	a^7
11	a^8
10	a^9
9	a^{10}
8	a^{11}
7	a^{12}
6	a^{13}
5	a^{14}
4	1
3	a
2	a^2
1	a^3

The configuration's list is

$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,1,2,2,2,2,2,2,2,2,2,2,$$
$$2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,$$
$$1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1$$

and the locator is

$$z_1 + a^2x_1^{14}x_2^7 + a^7x_1^{13}x_2^7 + ax_1^{12}x_2^7 + a^8x_1^{11}x_2^7 + x_1^{10}x_2^7 + x_1^9x_2^7 + x_1^8x_2^7 +$$
$$a^{13}x_1^7x_2^7 + x_1^6x_2^7 + a^{10}x_1^5x_2^7 + a^{11}x_1^4x_2^7 + a^4x_1^3x_2^7 + x_1^2x_2^7 + a^{14}x_1x_2^7 +$$
$$a^5x_2^7 + a^8x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^{13}x_1^{10}x_2^6 + x_1^9x_2^6 + a^{10}x_1^8x_2^6 + a^{11}x_1^7x_2^6 +$$
$$a^4x_1^6x_2^6 + x_1^5x_2^6 + a^{14}x_1^4x_2^6 + a^5x_1^3x_2^6 + a^2x_1^2x_2^6 + a^7x_1x_2^6 + ax_2^6 + a^{13}x_1^{13}x_2^5 +$$
$$a^{10}x_1^{11}x_2^5 + a^{11}x_1^{10}x_2^5 + a^4x_1^9x_2^5 + x_1^8x_2^5 + a^{14}x_1^7x_2^5 + a^5x_1^6x_2^5 + a^2x_1^5x_2^5 +$$
$$a^7x_1^4x_2^5 + ax_1^3x_2^5 + a^8x_1^2x_2^5 + a^{10}x_1x_2^5 + x_2^5 + a^4x_1^{12}x_2^4 + x_1^{11}x_2^4 + a^{14}x_1^{10}x_2^4 +$$
$$a^5x_1^9x_2^4 + a^2x_1^8x_2^4 + a^7x_1^7x_2^4 + ax_1^6x_2^4 + a^8x_1^5x_2^4 + a^9x_1^4x_2^4 + x_1^2x_2^4 + x_2^4 +$$
$$x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + x_1^3x_2^3 + a^2x_1^{10}x_2^2 + x_1^6x_2^2 + a^{10}x_1^5x_2^2 + a^{11}x_1^4x_2^2 + x_1^2x_2^2 +$$
$$a^{14}x_1x_2^2 + a^5x_2^2 + a^8x_1^{14}x_2 + a^{12}x_1^{13}x_2 + x_1^{12}x_2 + x_1^{11}x_2 + a^{13}x_1^{10}x_2 +$$
$$a^{10}x_1^8x_2 + a^4x_1^6x_2 + x_1^5x_2 + a^{14}x_1^4x_2 + a^2x_1^2x_2 + ax_2 + x_1^{14} + x_1^{12} + a^{10}x_1^{11} +$$
$$a^4x_1^9 + x_1^8 + a^5x_1^6 + a^2x_1^5 + ax_1^3 + a^8x_1^2 + a^{10}x_1 + 1$$

Configuration 14**:**

Number of points	Third coordinate
15	a^3
14	a^4
13	a^5
12	a^6
11	a^7
10	a^8
9	a^9
8	a^{10}
7	a^{11}
6	a^{12}
5	a^{13}
4	a^{14}
3	1
2	a
1	a^2

The configuration list is

$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,$$
$$2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1$$

and the locator is

$$z_1 + a^6x_1^{14}x_2^7 + a^{10}x_1^{13}x_2^7 + a^3x_1^{12}x_2^7 + a^9x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{14}x_1^9x_2^7 + a^{13}x_1^8x_2^7 + a^{10}x_1^7x_2^7 +$$
$$a^{11}x_1^6x_2^7 + a^5x_1^5x_2^7 + a^5x_1^4x_2^7 + a^{12}x_1^3x_2^7 + a^7x_1^2x_2^7 + a^5x_1x_2^7 + a^{10}x_2^7 + a^9x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 +$$
$$a^{10}x_1^{10}x_2^6 + a^{11}x_1^9x_2^6 + a^5x_1^8x_2^6 + a^5x_1^7x_2^6 + a^{12}x_1^6x_2^6 + a^7x_1^5x_2^6 + a^5x_1^4x_2^6 + a^{10}x_1^3x_2^6 + a^6x_1^2x_2^6 +$$
$$a^{10}x_1x_2^6 + a^3x_2^6 + a^{10}x_1^{13}x_2^5 + a^5x_1^{11}x_2^5 + a^5x_1^{10}x_2^5 + a^{12}x_1^9x_2^5 + a^7x_1^8x_2^5 + a^5x_1^7x_2^5 + a^{10}x_1^6x_2^5 +$$
$$a^6x_1^5x_2^5 + a^{10}x_1^4x_2^5 + a^3x_1^3x_2^5 + a^9x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{14}x_2^5 + a^{12}x_1^{12}x_2^4 + a^7x_1^{11}x_2^4 + a^5x_1^{10}x_2^4 +$$
$$a^{10}x_1^9x_2^4 + a^6x_1^8x_2^4 + a^{10}x_1^7x_2^4 + a^3x_1^6x_2^4 + a^9x_1^5x_2^4 + a^9x_1^4x_2^4 + a^{13}x_1^2x_2^4 + a^{11}x_2^4 + a^7x_1^{14}x_2^3 +$$
$$a^{10}x_1^7x_2^3 + a^{11}x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^{11}x_1^6x_2^2 + a^5x_1^5x_2^2 + a^5x_1^4x_2^2 + a^7x_1^2x_2^2 + a^5x_1x_2^2 + a^{10}x_2^2 +$$
$$a^9x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{14}x_1^{12}x_2 + a^{13}x_1^{11}x_2 + a^{10}x_1^{10}x_2 + a^5x_1^8x_2 + a^{12}x_1^6x_2 + a^7x_1^5x_2 +$$
$$a^5x_1^4x_2 + a^6x_1^2x_2 + a^3x_2 + a^{13}x_1^{14} + a^{11}x_1^{12} + a^5x_1^{11} + a^{12}x_1^9 + a^7x_1^8 + a^{10}x_1^6 + a^6x_1^5 +$$
$$a^3x_1^3 + a^9x_1^2 + a^{10}x_1 + a^{14}$$

Configuration 15**:**

Number of points	Third coordinate
15	a^2
14	a^3
13	a^4
12	a^5
11	a^6
10	a^7
9	a^8
8	a^9
7	a^{10}
6	a^{11}
5	a^{12}
4	a^{13}
3	a^{14}
2	1
1	a

The last configuration list is

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, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

and the locator is

$$z_1 + a^{10}x_1^{14}x_2^7 + a^{13}x_1^{13}x_2^7 + a^5x_1^{12}x_2^7 + a^{10}x_1^{11}x_2^7 + x_1^{10}x_2^7 + a^{13}x_1^9x_2^7 + a^{11}x_1^8x_2^7 + a^7x_1^7x_2^7 + a^7x_1^6x_2^7 + x_1^5x_2^7 + a^{14}x_1^4x_2^7 + a^5x_1^3x_2^7 + a^{14}x_1^2x_2^7 + a^{11}x_1x_2^7 + x_2^7 + a^{10}x_1^{14}x_2^6 + a^5x_1^{13}x_2^6 + a^7x_1^{10}x_2^6 + a^7x_1^9x_2^6 + x_1^8x_2^6 + a^{14}x_1^7x_2^6 + a^5x_1^6x_2^6 + a^{14}x_1^5x_2^6 + a^{11}x_1^4x_2^6 + x_1^3x_2^6 + a^{10}x_1^2x_2^6 + a^{13}x_1x_2^6 + a^5x_2^6 + a^7x_1^{13}x_2^5 + x_1^{11}x_2^5 + a^{14}x_1^{10}x_2^5 + a^5x_1^9x_2^5 + a^{14}x_1^8x_2^5 + a^{11}x_1^7x_2^5 + x_1^6x_2^5 + a^{10}x_1^5x_2^5 + a^{13}x_1^4x_2^5 + a^5x_1^3x_2^5 + a^{10}x_1^2x_2^5 + a^{10}x_1x_2^5 + a^{13}x_2^5 + a^5x_1^{12}x_2^4 + a^{14}x_1^{11}x_2^4 + a^{11}x_1^{10}x_2^4 + x_1^9x_2^4 + a^{10}x_1^8x_2^4 + a^{13}x_1^7x_2^4 + a^5x_1^6x_2^4 + a^{10}x_1^5x_2^4 + a^9x_1^4x_2^4 + a^{11}x_1^2x_2^4 + a^7x_2^4 + a^{14}x_1^{14}x_2^3 + a^{10}x_1^7x_2^3 + a^7x_1^3x_2^3 + a^2x_1^{10}x_2^2 + a^7x_1^6x_2^2 + x_1^5x_2^2 + a^{14}x_1^4x_2^2 + a^{14}x_1^2x_2^2 + a^{11}x_1x_2^2 + x_2^2 + a^{10}x_1^{14}x_2 + a^{12}x_1^{13}x_2 + a^{13}x_1^{12}x_2 + a^{11}x_1^{11}x_2 + a^7x_1^{10}x_2 + x_1^8x_2 + a^5x_1^6x_2 + a^{14}x_1^5x_2 + a^{11}x_1^4x_2 + a^{10}x_1^2x_2 + a^5x_2 + a^{11}x_1^{14} + a^7x_1^{12} + x_1^{11} + a^5x_1^9 + a^{14}x_1^8 + x_1^6 + a^{10}x_1^5 + a^5x_1^3 + a^{10}x_1^2 + a^{10}x_1 + a^{13}$$

B.2.2 Coefficient matrices and formulas.

The matrix whose entries are the terms which possibly can appear in the tail of our general error locator polynomial (8.4) is

$$M = \begin{bmatrix} x_1^{14} & x_1^{14}x_2 & x_1^{14}x_2^2 & x_1^{14}x_2^3 & x_1^{14}x_2^4 & x_1^{14}x_2^5 & x_1^{14}x_2^6 & x_1^{14}x_2^8 \\ x_1^{13} & x_1^{13}x_2 & x_1^{13}x_2^2 & x_1^{13}x_2^3 & x_1^{13}x_2^4 & x_1^{13}x_2^5 & x_1^{13}x_2^6 & x_1^{13}x_2^8 \\ x_1^{12} & x_1^{12}x_2 & x_1^{12}x_2^2 & x_1^{12}x_2^3 & x_1^{12}x_2^4 & x_1^{12}x_2^5 & x_1^{12}x_2^6 & x_1^{12}x_2^8 \\ x_1^{11} & x_1^{11}x_2 & x_1^{11}x_2^2 & x_1^{11}x_2^3 & x_1^{11}x_2^4 & x_1^{11}x_2^5 & x_1^{11}x_2^6 & x_1^{11}x_2^8 \\ x_1^{10} & x_1^{10}x_2 & x_1^{10}x_2^2 & x_1^{10}x_2^3 & x_1^{10}x_2^4 & x_1^{10}x_2^5 & x_1^{10}x_2^6 & x_1^{10}x_2^8 \\ x_1^9 & x_1^9x_2 & x_1^9x_2^2 & x_1^9x_2^3 & x_1^9x_2^4 & x_1^9x_2^5 & x_1^9x_2^6 & x_1^9x_2^8 \\ x_1^8 & x_1^8x_2 & x_1^8x_2^2 & x_1^8x_2^3 & x_1^8x_2^4 & x_1^8x_2^5 & x_1^8x_2^6 & x_1^8x_2^8 \\ x_1^7 & x_1^7x_2 & x_1^7x_2^2 & x_1^7x_2^3 & x_1^7x_2^4 & x_1^7x_2^5 & x_1^7x_2^6 & x_1^7x_2^8 \\ x_1^6 & x_1^6x_2 & x_1^6x_2^2 & x_1^6x_2^3 & x_1^6x_2^4 & x_1^6x_2^5 & x_1^6x_2^6 & x_1^6x_2^8 \\ x_1^5 & x_1^5x_2 & x_1^5x_2^2 & x_1^5x_2^3 & x_1^5x_2^4 & x_1^5x_2^5 & x_1^5x_2^6 & x_1^5x_2^8 \\ x_1^4 & x_1^4x_2 & x_1^4x_2^2 & x_1^4x_2^3 & x_1^4x_2^4 & x_1^4x_2^5 & x_1^4x_2^6 & x_1^4x_2^8 \\ x_1^3 & x_1^3x_2 & x_1^3x_2^2 & x_1^3x_2^3 & x_1^3x_2^4 & x_1^3x_2^5 & x_1^3x_2^6 & x_1^3x_2^8 \\ x_1^2 & x_1^2x_2 & x_1^2x_2^2 & x_1^2x_2^3 & x_1^2x_2^4 & x_1^2x_2^5 & x_1^2x_2^6 & x_1^2x_2^8 \\ x_1 & x_1x_2 & x_1x_2^2 & x_1x_2^3 & x_1x_2^4 & x_1x_2^5 & x_1x_2^6 & x_1x_2^8 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 & x_2^6 & x_2^8 \end{bmatrix} \tag{B.89}$$

The matrices of the fifteen cyclical configurations in $\mathbb{F}_{16}$ (8.4) are the following.

Configuration 1

$$A^{[1]} = \begin{pmatrix} a^9 & a^{11} & 0 & a^6 & 0 & 0 & a^{11} & a^{14} \\ 0 & a^{12} & 0 & 0 & 0 & a^4 & a^5 & a \\ a^3 & a^{12} & 0 & 0 & a^{13} & 0 & 0 & a^7 \\ a^{10} & a^9 & 0 & 0 & a^6 & a^{10} & 0 & a^{11} \\ 0 & a^4 & a^2 & 0 & a^2 & a^8 & a^4 & 1 \\ a^{13} & 0 & 0 & 0 & a^5 & a^{13} & a^3 & a^{12} \\ a^6 & a^{10} & 0 & 0 & a^{14} & a^6 & a^{10} & a^9 \\ 0 & 0 & 0 & a^{10} & a & a^2 & a^8 & a^4 \\ a^5 & a^{13} & a^3 & 0 & a^7 & a^5 & a^{13} & a^3 \\ a^{14} & a^6 & a^{10} & 0 & a^{11} & a^{14} & a^6 & a^{10} \\ 0 & a^2 & a^8 & 0 & a^9 & a & a^2 & a^8 \\ a^7 & 0 & 0 & a^3 & 0 & a^7 & a^5 & a^{13} \\ a^{11} & a^{14} & a^6 & 0 & a^9 & a^{11} & a^{14} & a^6 \\ a^{10} & 0 & a^2 & 0 & 0 & a^{10} & a & a^2 \\ a^{12} & a^7 & a^5 & 0 & a^3 & a^{12} & a^7 & a^5 \end{pmatrix}$$

Configuration 2

$$A^{[2]} = \begin{pmatrix} a^{11} & a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^{10} \\ 0 & a^{12} & 0 & 0 & 0 & a^7 & a^5 & a^{13} \\ a^7 & a^{13} & 0 & 0 & a^5 & 0 & 0 & a^5 \\ 1 & a^{11} & 0 & 0 & a^{14} & 1 & 0 & a^{10} \\ 0 & a^7 & a^2 & 0 & a^{11} & a^{14} & a^7 & 1 \\ a^5 & 0 & 0 & 0 & 1 & a^5 & a^7 & a^{13} \\ a^{14} & 1 & 0 & 0 & a^{10} & a^{14} & 1 & a^{11} \\ 0 & 0 & 0 & a^{10} & a^{13} & a^{11} & a^{14} & a^7 \\ 1 & a^5 & a^7 & 0 & a^5 & 1 & a^5 & a^7 \\ a^{10} & a^{14} & 1 & 0 & a^{10} & a^{10} & a^{14} & 1 \\ 0 & a^{11} & a^{14} & 0 & a^9 & a^{13} & a^{11} & a^{14} \\ a^5 & 0 & 0 & a^7 & 0 & a^5 & 1 & a^5 \\ a^{10} & a^{10} & a^{14} & 0 & a^{11} & a^{10} & a^{10} & a^{14} \\ a^{10} & 0 & a^{11} & 0 & 0 & a^{10} & a^{13} & a^{11} \\ a^{13} & a^5 & 1 & 0 & a^7 & a^{13} & a^5 & 1 \end{pmatrix}$$

Configuration 3

$$A^{[3]} = \begin{pmatrix} a^5 & a^{13} & 0 & a^5 & 0 & 0 & a^{13} & a^7 \\ 0 & a^{12} & 0 & 0 & 0 & a^{13} & a^5 & a^7 \\ a^{10} & a^{10} & 0 & 0 & a^{14} & 0 & 0 & a^{11} \\ 1 & a^5 & 0 & 0 & a^5 & 1 & 0 & a^{13} \\ 0 & a^{13} & a^2 & 0 & a^{14} & a^{11} & a^{13} & 1 \\ a^{14} & 0 & 0 & 0 & 1 & a^{14} & a^{10} & a^{10} \\ a^5 & 1 & 0 & 0 & a^7 & a^5 & 1 & a^5 \\ 0 & 0 & 0 & a^{10} & a^7 & a^{14} & a^{11} & a^{13} \\ 1 & a^{14} & a^{10} & 0 & a^{11} & 1 & a^{14} & a^{10} \\ a^7 & a^5 & 1 & 0 & a^{13} & a^7 & a^5 & 1 \\ 0 & a^{14} & a^{11} & 0 & a^9 & a^7 & a^{14} & a^{11} \\ a^{11} & 0 & 0 & a^{10} & 0 & a^{11} & 1 & a^{14} \\ a^{13} & a^7 & a^5 & 0 & a^5 & a^{13} & a^7 & a^5 \\ a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^7 & a^{14} \\ a^{10} & a^{11} & 1 & 0 & a^{10} & a^{10} & a^{11} & 1 \end{pmatrix}$$

Configuration 4

$$A^{[4]} = \begin{pmatrix} a^3 & a^{14} & 0 & a^{12} & 0 & 0 & a^{14} & a^{11} \\ 0 & a^{12} & 0 & 0 & 0 & a^{10} & a^5 & a^{10} \\ a^6 & a^9 & 0 & 0 & a^7 & 0 & 0 & a^{13} \\ a^{10} & a^3 & 0 & 0 & a^{12} & a^{10} & 0 & a^{14} \\ 0 & a^{10} & a^2 & 0 & a^5 & a^5 & a^{10} & 1 \\ a^7 & 0 & 0 & 0 & a^5 & a^7 & a^6 & a^9 \\ a^{12} & a^{10} & 0 & 0 & a^{11} & a^{12} & a^{10} & a^3 \\ 0 & 0 & 0 & a^{10} & a^{10} & a^5 & a^5 & a^{10} \\ a^5 & a^7 & a^6 & 0 & a^{13} & a^5 & a^7 & a^6 \\ a^{11} & a^{12} & a^{10} & 0 & a^{14} & a^{11} & a^{12} & a^{10} \\ 0 & a^5 & a^5 & 0 & a^9 & a^{10} & a^5 & a^5 \\ a^{13} & 0 & 0 & a^6 & 0 & a^{13} & a^5 & a^7 \\ a^{14} & a^{11} & a^{12} & 0 & a^3 & a^{14} & a^{11} & a^{12} \\ a^{10} & 0 & a^5 & 0 & 0 & a^{10} & a^{10} & a^5 \\ a^9 & a^{13} & a^5 & 0 & a^6 & a^9 & a^{13} & a^5 \end{pmatrix}$$

Configuration 5

$$A^{[5]} = \begin{pmatrix} a & 1 & 0 & a^4 & 0 & 0 & 1 & 1 \\ 0 & a^{12} & 0 & 0 & 0 & a^7 & a^5 & a^{13} \\ a^2 & a^8 & 0 & 0 & 1 & 0 & 0 & 1 \\ a^5 & a & 0 & 0 & a^4 & a^5 & 0 & 1 \\ 0 & a^7 & a^2 & 0 & a^{11} & a^{14} & a^7 & 1 \\ 1 & 0 & 0 & 0 & a^{10} & 1 & a^2 & a^8 \\ a^4 & a^5 & 0 & 0 & 1 & a^4 & a^5 & a \\ 0 & 0 & 0 & a^{10} & a^{13} & a^{11} & a^{14} & a^7 \\ a^{10} & 1 & a^2 & 0 & 1 & a^{10} & 1 & a^2 \\ 1 & a^4 & a^5 & 0 & 1 & 1 & a^4 & a^5 \\ 0 & a^{11} & a^{14} & 0 & a^9 & a^{13} & a^{11} & a^{14} \\ 1 & 0 & 0 & a^2 & 0 & 1 & a^{10} & 1 \\ 1 & 1 & a^4 & 0 & a & 1 & 1 & a^4 \\ a^{10} & 0 & a^{11} & 0 & 0 & a^{10} & a^{13} & a^{11} \\ a^8 & 1 & a^{10} & 0 & a^2 & a^8 & 1 & a^{10} \end{pmatrix}$$

Configuration 6

$$A^{[6]} = \begin{pmatrix} a^{14} & a & 0 & a^{11} & 0 & 0 & a & a^4 \\ 0 & a^{12} & 0 & 0 & 0 & a^4 & a^5 & a \\ a^{13} & a^7 & 0 & 0 & a^8 & 0 & 0 & a^2 \\ 1 & a^{14} & 0 & 0 & a^{11} & 1 & 0 & a \\ 0 & a^4 & a^2 & 0 & a^2 & a^8 & a^4 & 1 \\ a^8 & 0 & 0 & 0 & 1 & a^8 & a^{13} & a^7 \\ a^{11} & 1 & 0 & 0 & a^4 & a^{11} & 1 & a^{14} \\ 0 & 0 & 0 & a^{10} & a & a^2 & a^8 & a^4 \\ 1 & a^8 & a^{13} & 0 & a^2 & 1 & a^8 & a^{13} \\ a^4 & a^{11} & 1 & 0 & a & a^4 & a^{11} & 1 \\ 0 & a^2 & a^8 & 0 & a^9 & a & a^2 & a^8 \\ a^2 & 0 & 0 & a^{13} & 0 & a^2 & 1 & a^8 \\ a & a^4 & a^{11} & 0 & a^{14} & a & a^4 & a^{11} \\ a^{10} & 0 & a^2 & 0 & 0 & a^{10} & a & a^2 \\ a^7 & a^2 & 1 & 0 & a^{13} & a^7 & a^2 & 1 \end{pmatrix}$$

Configuration 7

$$A^{[7]} = \begin{pmatrix} a^{12} & a^2 & 0 & a^3 & 0 & 0 & a^2 & a^8 \\ 0 & a^{12} & 0 & 0 & 0 & a & a^5 & a^4 \\ a^9 & a^6 & 0 & 0 & a & 0 & 0 & a^4 \\ a^{10} & a^{12} & 0 & 0 & a^3 & a^{10} & 0 & a^2 \\ 0 & a & a^2 & 0 & a^8 & a^2 & a & 1 \\ a & 0 & 0 & 0 & a^5 & a & a^9 & a^6 \\ a^3 & a^{10} & 0 & 0 & a^8 & a^3 & a^{10} & a^{12} \\ 0 & 0 & 0 & a^{10} & a^4 & a^8 & a^2 & a \\ a^5 & a & a^9 & 0 & a^4 & a^5 & a & a^9 \\ a^8 & a^3 & a^{10} & 0 & a^2 & a^8 & a^3 & a^{10} \\ 0 & a^8 & a^2 & 0 & a^9 & a^4 & a^8 & a^2 \\ a^4 & 0 & 0 & a^9 & 0 & a^4 & a^5 & a \\ a^2 & a^8 & a^3 & 0 & a^{12} & a^2 & a^8 & a^3 \\ a^{10} & 0 & a^8 & 0 & 0 & a^{10} & a^4 & a^8 \\ a^6 & a^4 & a^5 & 0 & a^9 & a^6 & a^4 & a^5 \end{pmatrix}$$

Configuration 8

$$A^{[8]} = \begin{pmatrix} a^{10} & a^3 & 0 & a^{10} & 0 & 0 & a^3 & a^{12} \\ 0 & a^{12} & 0 & 0 & 0 & a^{13} & a^5 & a^7 \\ a^5 & a^5 & 0 & 0 & a^9 & 0 & 0 & a^6 \\ a^5 & a^{10} & 0 & 0 & a^{10} & a^5 & 0 & a^3 \\ 0 & a^{13} & a^2 & 0 & a^{14} & a^{11} & a^{13} & 1 \\ a^9 & 0 & 0 & 0 & a^{10} & a^9 & a^5 & a^5 \\ a^{10} & a^5 & 0 & 0 & a^{12} & a^{10} & a^5 & a^{10} \\ 0 & 0 & 0 & a^{10} & a^7 & a^{14} & a^{11} & a^{13} \\ a^{10} & a^9 & a^5 & 0 & a^6 & a^{10} & a^9 & a^5 \\ a^{12} & a^{10} & a^5 & 0 & a^3 & a^{12} & a^{10} & a^5 \\ 0 & a^{14} & a^{11} & 0 & a^9 & a^7 & a^{14} & a^{11} \\ a^6 & 0 & 0 & a^5 & 0 & a^6 & a^{10} & a^9 \\ a^3 & a^{12} & a^{10} & 0 & a^{10} & a^3 & a^{12} & a^{10} \\ a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^7 & a^{14} \\ a^5 & a^6 & a^{10} & 0 & a^5 & a^5 & a^6 & a^{10} \end{pmatrix}$$

Configuration 9

$$A^{[9]} = \begin{pmatrix} a^8 & a^4 & 0 & a^2 & 0 & 0 & a^4 & a \\ 0 & a^{12} & 0 & 0 & 0 & a^{10} & a^5 & a^{10} \\ a & a^4 & 0 & 0 & a^2 & 0 & 0 & a^8 \\ 1 & a^8 & 0 & 0 & a^2 & 1 & 0 & a^4 \\ 0 & a^{10} & a^2 & 0 & a^5 & a^5 & a^{10} & 1 \\ a^2 & 0 & 0 & 0 & 1 & a^2 & a & a^4 \\ a^2 & 1 & 0 & 0 & a & a^2 & 1 & a^8 \\ 0 & 0 & 0 & a^{10} & a^{10} & a^5 & a^5 & a^{10} \\ 1 & a^2 & a & 0 & a^8 & 1 & a^2 & a \\ a & a^2 & 1 & 0 & a^4 & a & a^2 & 1 \\ 0 & a^5 & a^5 & 0 & a^9 & a^{10} & a^5 & a^5 \\ a^8 & 0 & 0 & a & 0 & a^8 & 1 & a^2 \\ a^4 & a & a^2 & 0 & a^8 & a^4 & a & a^2 \\ a^{10} & 0 & a^5 & 0 & 0 & a^{10} & a^{10} & a^5 \\ a^4 & a^8 & 1 & 0 & a & a^4 & a^8 & 1 \end{pmatrix}$$

Configuration 10

$$A^{[10]} = \begin{pmatrix} a^6 & a^5 & 0 & a^9 & 0 & 0 & a^5 & a^5 \\ 0 & a^{12} & 0 & 0 & 0 & a^7 & a^5 & a^{13} \\ a^{12} & a^3 & 0 & 0 & a^{10} & 0 & 0 & a^{10} \\ a^{10} & a^6 & 0 & 0 & a^9 & a^{10} & 0 & a^5 \\ 0 & a^7 & a^2 & 0 & a^{11} & a^{14} & a^7 & 1 \\ a^{10} & 0 & 0 & 0 & a^5 & a^{10} & a^{12} & a^3 \\ a^9 & a^{10} & 0 & 0 & a^5 & a^9 & a^{10} & a^6 \\ 0 & 0 & 0 & a^{10} & a^{13} & a^{11} & a^{14} & a^7 \\ a^5 & a^{10} & a^{12} & 0 & a^{10} & a^5 & a^{10} & a^{12} \\ a^5 & a^9 & a^{10} & 0 & a^5 & a^5 & a^9 & a^{10} \\ 0 & a^{11} & a^{14} & 0 & a^9 & a^{13} & a^{11} & a^{14} \\ a^{10} & 0 & 0 & a^{12} & 0 & a^{10} & a^5 & a^{10} \\ a^5 & a^5 & a^9 & 0 & a^6 & a^5 & a^5 & a^9 \\ a^{10} & 0 & a^{11} & 0 & 0 & a^{10} & a^{13} & a^{11} \\ a^3 & a^{10} & a^5 & 0 & a^{12} & a^3 & a^{10} & a^5 \end{pmatrix}$$

Configuration 11

$$A^{[11]} = \begin{pmatrix} a^4 & a^6 & 0 & a & 0 & 0 & a^6 & a^9 \\ 0 & a^{12} & 0 & 0 & 0 & a^4 & a^5 & a \\ a^8 & a^2 & 0 & 0 & a^3 & 0 & 0 & a^{12} \\ a^5 & a^4 & 0 & 0 & a & a^5 & 0 & a^6 \\ 0 & a^4 & a^2 & 0 & a^2 & a^8 & a^4 & 1 \\ a^3 & 0 & 0 & 0 & a^{10} & a^3 & a^8 & a^2 \\ a & a^5 & 0 & 0 & a^9 & a & a^5 & a^4 \\ 0 & 0 & 0 & a^{10} & a & a^2 & a^8 & a^4 \\ a^{10} & a^3 & a^8 & 0 & a^{12} & a^{10} & a^3 & a^8 \\ a^9 & a & a^5 & 0 & a^6 & a^9 & a & a^5 \\ 0 & a^2 & a^8 & 0 & a^9 & a & a^2 & a^8 \\ a^{12} & 0 & 0 & a^8 & 0 & a^{12} & a^{10} & a^3 \\ a^6 & a^9 & a & 0 & a^4 & a^6 & a^9 & a \\ a^{10} & 0 & a^2 & 0 & 0 & a^{10} & a & a^2 \\ a^2 & a^{12} & a^{10} & 0 & a^8 & a^2 & a^{12} & a^{10} \end{pmatrix}$$

Configuration 12

$$A^{[12]} = \begin{pmatrix} a^2 & a^7 & 0 & a^8 & 0 & 0 & a^7 & a^{13} \\ 0 & a^{12} & 0 & 0 & 0 & a & a^5 & a^4 \\ a^4 & a & 0 & 0 & a^{11} & 0 & 0 & a^{14} \\ 1 & a^2 & 0 & 0 & a^8 & 1 & 0 & a^7 \\ 0 & a & a^2 & 0 & a^8 & a^2 & a & 1 \\ a^{11} & 0 & 0 & 0 & 1 & a^{11} & a^4 & a \\ a^8 & 1 & 0 & 0 & a^{13} & a^8 & 1 & a^2 \\ 0 & 0 & 0 & a^{10} & a^4 & a^8 & a^2 & a \\ 1 & a^{11} & a^4 & 0 & a^{14} & 1 & a^{11} & a^4 \\ a^{13} & a^8 & 1 & 0 & a^7 & a^{13} & a^8 & 1 \\ 0 & a^8 & a^2 & 0 & a^9 & a^4 & a^8 & a^2 \\ a^{14} & 0 & 0 & a^4 & 0 & a^{14} & 1 & a^{11} \\ a^7 & a^{13} & a^8 & 0 & a^2 & a^7 & a^{13} & a^8 \\ a^{10} & 0 & a^8 & 0 & 0 & a^{10} & a^4 & a^8 \\ a & a^{14} & 1 & 0 & a^4 & a & a^{14} & 1 \end{pmatrix}$$

Configuration 13

$$A^{[13]} = \begin{pmatrix} 1 & a^{8} & 0 & 1 & 0 & 0 & a^{8} & a^{2} \\ 0 & a^{12} & 0 & 0 & 0 & a^{13} & a^{5} & a^{7} \\ 1 & 1 & 0 & 0 & a^{4} & 0 & 0 & a \\ a^{10} & 1 & 0 & 0 & 1 & a^{10} & 0 & a^{8} \\ 0 & a^{13} & a^{2} & 0 & a^{14} & a^{11} & a^{13} & 1 \\ a^{4} & 0 & 0 & 0 & a^{5} & a^{4} & 1 & 1 \\ 1 & a^{10} & 0 & 0 & a^{2} & 1 & a^{10} & 1 \\ 0 & 0 & 0 & a^{10} & a^{7} & a^{14} & a^{11} & a^{13} \\ a^{5} & a^{4} & 1 & 0 & a & a^{5} & a^{4} & 1 \\ a^{2} & 1 & a^{10} & 0 & a^{8} & a^{2} & 1 & a^{10} \\ 0 & a^{14} & a^{11} & 0 & a^{9} & a^{7} & a^{14} & a^{11} \\ a & 0 & 0 & 1 & 0 & a & a^{5} & a^{4} \\ a^{8} & a^{2} & 1 & 0 & 1 & a^{8} & a^{2} & 1 \\ a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^{7} & a^{14} \\ 1 & a & a^{5} & 0 & 1 & 1 & a & a^{5} \end{pmatrix}$$

Configuration 14

$$A^{[14]} = \begin{pmatrix} a^{13} & a^{9} & 0 & a^{7} & 0 & 0 & a^{9} & a^{6} \\ 0 & a^{12} & 0 & 0 & 0 & a^{10} & a^{5} & a^{10} \\ a^{11} & a^{14} & 0 & 0 & a^{12} & 0 & 0 & a^{3} \\ a^{5} & a^{13} & 0 & 0 & a^{7} & a^{5} & 0 & a^{9} \\ 0 & a^{10} & a^{2} & 0 & a^{5} & a^{5} & a^{10} & 1 \\ a^{12} & 0 & 0 & 0 & a^{10} & a^{12} & a^{11} & a^{14} \\ a^{7} & a^{5} & 0 & 0 & a^{6} & a^{7} & a^{5} & a^{13} \\ 0 & 0 & 0 & a^{10} & a^{10} & a^{5} & a^{5} & a^{10} \\ a^{10} & a^{12} & a^{11} & 0 & a^{3} & a^{10} & a^{12} & a^{11} \\ a^{6} & a^{7} & a^{5} & 0 & a^{9} & a^{6} & a^{7} & a^{5} \\ 0 & a^{5} & a^{5} & 0 & a^{9} & a^{10} & a^{5} & a^{5} \\ a^{3} & 0 & 0 & a^{11} & 0 & a^{3} & a^{10} & a^{12} \\ a^{9} & a^{6} & a^{7} & 0 & a^{13} & a^{9} & a^{6} & a^{7} \\ a^{10} & 0 & a^{5} & 0 & 0 & a^{10} & a^{10} & a^{5} \\ a^{14} & a^{3} & a^{10} & 0 & a^{11} & a^{14} & a^{3} & a^{10} \end{pmatrix}$$

Configuration 15

$$A^{[15]} = \begin{pmatrix} a^{11} & a^{10} & 0 & a^{14} & 0 & 0 & a^{10} & a^{10} \\ 0 & a^{12} & 0 & 0 & 0 & a^{7} & a^{5} & a^{13} \\ a^{7} & a^{13} & 0 & 0 & a^{5} & 0 & 0 & a^{5} \\ 1 & a^{11} & 0 & 0 & a^{14} & 1 & 0 & a^{10} \\ 0 & a^{7} & a^{2} & 0 & a^{11} & a^{14} & a^{7} & 1 \\ a^{5} & 0 & 0 & 0 & 1 & a^{5} & a^{7} & a^{13} \\ a^{14} & 1 & 0 & 0 & a^{10} & a^{14} & 1 & a^{11} \\ 0 & 0 & 0 & a^{10} & a^{13} & a^{11} & a^{14} & a^{7} \\ 1 & a^{5} & a^{7} & 0 & a^{5} & 1 & a^{5} & a^{7} \\ a^{10} & a^{14} & 1 & 0 & a^{10} & a^{10} & a^{14} & 1 \\ 0 & a^{11} & a^{14} & 0 & a^{9} & a^{13} & a^{11} & a^{14} \\ a^{5} & 0 & 0 & a^{7} & 0 & a^{5} & 1 & a^{5} \\ a^{10} & a^{10} & a^{14} & 0 & a^{11} & a^{10} & a^{10} & a^{14} \\ a^{10} & 0 & a^{11} & 0 & 0 & a^{10} & a^{13} & a^{11} \\ a^{13} & a^{5} & 1 & 0 & a^{7} & a^{13} & a^{5} & 1 \end{pmatrix}$$

As for the intermediate configurations in $\mathbb{F}_8$ we can find a general table, summarizing the reciprocal relations among the coefficients of each locator polynomial:

$$A^{[\text{gen}]} \begin{pmatrix} B & A & 0 & C & 0 & 0 & A & D \\ 0 & a^{12} & 0 & 0 & 0 & E & a^{5} & F \\ G & H & 0 & 0 & I & 0 & 0 & L \\ M & B & 0 & 0 & C & M & 0 & A \\ 0 & E & a^{2} & 0 & N & O & E & 1 \\ I & 0 & 0 & 0 & P & I & G & H \\ C & M & 0 & 0 & D & C & M & B \\ 0 & 0 & 0 & a^{10} & F & N & O & E \\ P & I & G & 0 & L & P & I & G \\ D & C & M & 0 & A & D & C & M \\ 0 & N & O & 0 & a^{9} & F & N & O \\ L & 0 & 0 & G & 0 & L & P & I \\ A & D & C & 0 & B & A & D & C \\ a^{10} & 0 & N & 0 & 0 & a^{10} & F & N \\ H & L & P & 0 & G & H & L & P \end{pmatrix}.$$

The numbers $1, ..., 15$ of the first row represent the number of occurrences of the value we take as Q (B.2.2).

A-P	$Q^?$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	Q^{14}	12	13	14	15	1	2	3	4	5	6	7	8	9	10	11
B	Q^2	7	5	3	1	14	12	10	8	6	4	2	15	13	11	9
C	Q^8	13	5	12	4	11	3	10	2	9	1	8	15	7	14	6
D	Q^{11}	3	7	11	15	4	8	12	1	5	9	13	2	6	10	14
E	Q^3	1	13	10	7	4	1	13	10	7	4	1	13	10	7	4
F	Q^{12}	4	7	10	13	1	4	7	10	13	1	4	7	10	13	1
G	Q^4	14	10	6	2	13	9	5	1	12	8	4	15	11	7	3
H	Q	11	10	9	8	7	6	5	4	3	2	1	15	14	13	12
I	Q^7	6	14	7	15	8	1	9	2	10	3	11	4	12	5	13
L	Q^{13}	9	11	13	15	2	4	6	8	10	12	14	1	3	5	7
Q	Q^5	5	15	10	5	15	10	5	15	10	5	15	10	5	15	10
N	Q^9	8	14	5	11	2	8	14	5	11	2	8	14	5	11	2
O	Q^6	2	11	5	14	8	2	11	5	14	8	2	11	5	14	8
P	Q^{10}	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5

Table B.2: Configurations in $\mathbb{F}_{16}$.

The couples of cycles corresponding to the formulas grouped above are:

1: (α', δ'), (β', α'), (γ', γ'), (δ', β');

2: (α', β'), (β', α'), (γ', ϵ'), (δ', δ');

3: (α', β'), (β', γ'), (γ', γ'), (δ', δ');

4: (α', α'), (β', δ'), (γ', γ'), (δ', γ');

5: (α', δ'), (β', α'), (γ', ϵ'), (δ', α');

6: (α', β'), (β', α'), (γ', γ'), (δ', α');

7: (α', γ'), (β', δ'), (γ', γ'), (δ', β');

8: (α', α'), (β', δ'), (γ', ϵ'), (δ', α');

9: (α', β'), (β', δ'), (γ', γ'), (δ', γ');

10: (α', α'), (β', α'), (γ', γ'), (δ', β');

11: (α', α'), (β', α'), (γ', ϵ'), (δ', δ');

12: (α', ϵ'), (β', δ'), (γ', γ'), (δ', α');

13: (α', δ'), (β', γ'), (γ', γ'), (δ', β');

14: (α', δ'), (β', δ'), (γ', ϵ'), (δ', γ');

15: (α', β'), (β', α'), (γ', γ'), (δ', δ').

B.3 Optimal Frobenius configurations in $\mathbb{F}_8$.

In the case of $\mathbb{F}_8$ we were able to find some optimal Frobenius configurations and some optimal semi-Frobenius configurations.
We arranged these configurations in type A, B, C, D.
We display here all the precise data for them.

B.3.1 Nine terms

We give here the data for the nine term polynomial leading to the optimal Frobenius configurations (8.5) and its failing permutations.
Let us start with the nine therm configuration:

Number of points	Third coordinate
7	0
6	a
3	a^2
3	a^3
3	a^5
2	a^4
2	a^6
2	1

The points are:

$$[(a, a^3, 0, a)],$$
$$[(a^2, a^6, 0, a^2)],$$
$$[(a^3, a^2, 0, a^3)],$$
$$[(a^4, a^5, 0, a^4)],$$
$$[(a^5, a, 0, a^5)],$$
$$[(a^6, a^4, 0, a^6)],$$
$$[(1, 1, 0, 1)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a, a^4)],$$
$$[(a^6, 1, a, a^5)],$$
$$[(a^5, a^6, a, a^6)],$$
$$[(a^3, a, a, 1)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^4, a^2)],$$
$$[(a^3, a^5, a^5, a^2)],$$
$$[(1, a^3, a^2, a^6)],$$
$$[(a^6, a^2, a^2, 1)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^5, a^4)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^6, a^5)],$$
$$[(a^4, a^3, a^5, 1)],$$
$$[(a^2, a^5, a^6, 1)].$$

The configuration list is

$$[1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,1,1,1,1,1,2,2,1,2,2,1,1]$$

and the associated tower structure is

a,a^2,a^6	a^2,a^5,a^6	a^3,1,a^4	a^4,a^3,a^5	a^5,a^4,1	a^6,a^3,a^3	1,a^6,a^5
a,a^6,1	a^2,a^4,a^3	a^3,a^5,a^5	a^4,a,a^3	a^5,1,a^2	a^6,a^2,a^2	1,a^3,a^2
a,a,a^4	a^2,a^2,a	a^3,a,a	a^4,a^4,a	a^5,a^6,a	a^6,1,a	1,a^5,a
a,a^3,0	a^2,a^6,0	a^3,a^2,0	a^4,a^5,0	a^5,a,0	a^6,a^4,0	1,1,0

The locator polynomial we get is:
$z_1 + x_1^6 x_2^3 + a^3 x_1^6 x_2^2 + a^5 x_1^4 x_2^2 + a^6 x_1^6 x_2 + a^3 x_1^2 x_2 + a^5 x_1^3 + a^6 x_1^2 + x_1$
Let us try now to variate it.

Number of points	Third coordinate
7	1
6	0
3	a
3	a^2
3	a^3
2	a^5
2	a^4
2	a^6

The points are:

$$[(a, a^3, 0, a)],$$
$$[(a^2, a^6, 0, a^2)],$$
$$[(a^3, a^2, 0, a^3)],$$
$$[(a^4, a^5, 0, a^4)],$$
$$[(a^5, a, 0, a^5)],$$
$$[(a^6, a^4, 0, a^6)],$$
$$[(1, 1, 1, 0)],$$

$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a, a^4)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$
$$[(a^3, a^5, a^2, a^5)],$$
$$[(1, a^3, a^6, a^2)],$$
$$[(a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, 1, a^5)],$$
$$[(a^2, a^5, 1, a^6)].$$

corresponding to

$$[1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1, 1, 2, 1, 2, 2].$$

and to

a,a^2,a^5	a^2,a^5,1	a^3,1,a^4	a^4,a^3,1	a^5,a^4,1	a^6,a^3,a^3	1,a^6,a^4
a,a^6,1	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,1	1,a^3,a^6
a,a,a^2	a^2,a^2,a	a^3,a,1	a^4,a^4,a	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a
a,a^3,0	a^2,a^6,0	a^3,a^2,0	a^4,a^5,0	a^5,a,0	a^6,a^4,0	1,1,1

The polynomial we get is:
$z_1 + x_1^6x_2^3 + a^2x_1^5x_2^3 + a^4x_1^4x_2^3 + x_1^3x_2^3 + ax_1^2x_2^3 + x_1x_2^3 + x_2^3 + a^3x_1^6x_2^2 + a^6x_1^5x_2^2 + ax_1^4x_2^2 + ax_1^3x_2^2 +$

$ax_1^2x_2^2 + a^6x_1x_2^2 + ax_2^2 + a^2x_1^6x_2 + a^2x_1^5x_2 + a^3x_1^4x_2 + x_1^3x_2 + a^6x_1^2x_2 + a^4x_1x_2 + a^6x_2 + a^2x_1^6 + a^4x_1^5 + a^5x_1^3 + a^4x_1^2 + a^4x_1 + a^3$.

In analogy with the intermediate configuration, we try to remove the possibility for zero to be the third coordinate.

Number of points	Third coordinate
7	0
6	1
3	a
3	a^2
3	a^3
2	a^5
2	a^4
2	a^6

The points are:

$$[(a, a^3, 0, a)],$$
$$[(a^2, a^6, 0, a^2)],$$
$$[(a^3, a^2, 0, a^3)],$$
$$[(a^4, a^5, 0, a^4)],$$
$$[(a^5, a, 0, a^5)],$$
$$[(a^6, a^4, 0, a^6)],$$
$$[(1, 1, 0, 1)],$$
$$[(a^4, a^4, a, a^2)],$$
$$[(1, a^5, a, a^3)],$$
$$[(a^2, a^2, a, a^4)],$$
$$[(a^6, 1, a^5, a)],$$
$$[(a^5, a^6, a^6, a)],$$
$$[(a^3, a, 1, a)],$$
$$[(a^5, 1, a^2, a^3)],$$
$$[(a, a, a^2, a^4)],$$

$$[(a^3, a^5, a^2, a^5)],$$
$$[(1, a^3, a^6, a^2)],$$
$$[(a^6, a^2, 1, a^2)],$$
$$[(a^6, a^3, a^3, a^4)],$$
$$[(a^2, a^4, a^3, a^5)],$$
$$[(a^4, a, a^3, a^6)],$$
$$[(a, a^6, 1, a^3)],$$
$$[(1, a^6, a^4, a^5)],$$
$$[(a^3, 1, a^4, a^6)],$$
$$[(a^5, a^4, 1, a^4)],$$
$$[(a, a^2, a^5, a^6)],$$
$$[(a^4, a^3, 1, a^5)],$$
$$[(a^2, a^5, 1, a^6)].$$

corresponding to

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 2]$$

and to

a,a^2,a^5	a^2,a^5,1	a^3,1,a^4	a^4,a^3,1	a^5,a^4,1	a^6,a^3,a^3	1,a^6,a^4
a,a^6,1	a^2,a^4,a^3	a^3,a^5,a^2	a^4,a,a^3	a^5,1,a^2	a^6,a^2,1	1,a^3,a^6
a,a,a^2	a^2,a^2,a	a^3,a,1	a^4,a^4,a	a^5,a^6,a^6	a^6,1,a^5	1,a^5,a
a,a^3,0	a^2,a^6,0	a^3,a^2,0	a^4,a^5,0	a^5,a,0	a^6,a^4,0	1,1,0

The obtained polynomial is made up of 28 terms:
$z_1 + a^6x_1^5x_2^3 + a^5x_1^4x_2^3 + x_1^3x_2^3 + a^3x_1^2x_2^3 + x_1x_2^3 + x_2^3 + ax_1^6x_2^2 + a^5x_1^5x_2^2 + a^2x_1^4x_2^2 + a^5x_1^3x_2^2 + a^5x_1^2x_2^2 + a^2x_1x_2^2 + a^2x_2^2 + a^6x_1^6x_2 + ax_1^5x_2 + a^4x_1^4x_2 + x_1^3x_2 + a^5x_1^2x_2 + a^6x_1x_2 + ax_2 + ax_1^6 + a^2x_1^5 + x_1^4 + a^5x_1^3 + a^3x_1^2 + a^6x_1 + a^5$

B.3.2 Type A.

Here we have all the data for Type A configurations from 8.5.

Type A:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

The first column is associated to the following configuration list:
$2,2,2,2,2,2,2,2,2,2,2,2,2,1,2,2,1,2,1,1,2,2,1,2,1,2,1,2.$
The associated polynomial is
$z_1 + a^3x_1^6x_2^2 + a^6x_1^3x_2^2 + x_1^2x_2^2 + a^6x_1^6x_2 + a^5x_2 + a^3x_1^5 + a^5x_1^3$
and the matrix of coefficients (completely analogous to table 8.1) turns out to be:

$$A^{[1]} = \begin{pmatrix} 0 & a^6 & a^3 & 0 \\ a^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^5 & 0 & a^6 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^5 & 0 & 0 \end{pmatrix}$$

The second column corresponds to the list below:
$2,2,2,2,2,2,2,1,2,2,1,2,1,1,1,2,2,1,1,2,1,1,2,1,1,2,1,1$ and to the locator polynomial
$z_1 + x_1^6x_2^2 + x_1^3x_2^2 + x_1^2x_2^2 + x_1^6x_2 + x_2 + x_1^5 + x_1^3$

The coefficient table is

$$A^{[2]} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

The third column gives the following list:
$2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 2$ and the locator polynomial is $z_1 + a^4x_1^6x_2^2 + ax_1^3x_2^2 + x_1^2x_2^2 + ax_1^6x_2 + a^2x_2 + a^4x_1^5 + a^2x_1^3$
while the coefficient table is

$$A^{[3]} = \begin{pmatrix} 0 & a & a^4 & 0 \\ a^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^2 & 0 & a & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^2 & 0 & 0 \end{pmatrix}$$

The fourth column is associated to
$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 2, 2, 1$
The locator is $z_1 + ax_1^6x_2^2 + a^2x_1^3x_2^2 + x_1^2x_2^2 + a^2x_1^6x_2 + a^4x_2 + ax_1^5 + a^4x_1^3$
and the table

$$A^{[4]} = \begin{pmatrix} 0 & a^2 & a & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^4 & 0 & a^2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^4 & 0 & 0 \end{pmatrix}$$

The configuration list associated to the fifth column is
$2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1$ and the locator polynomial is $z_1 + a^5x_1^6x_2^2 + a^3x_1^3x_2^2 + x_1^2x_2^2 + a^3x_1^6x_2 + a^6x_2 + a^5x_1^5 + a^6x_1^3$.

The table grouping the coefficients is

$$A^{[5]} = \begin{pmatrix} 0 & a^3 & a^5 & 0 \\ a^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^6 & 0 & a^3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^6 & 0 & 0 \end{pmatrix}$$

The sixth column is connected to
$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1$ and to
$z_1 + a^2x_1^6x_2^2 + a^4x_1^3x_2^2 + x_1^2x_2^2 + a^4x_1^6x_2 + ax_2 + a^2x_1^5 + ax_1^3$
The table is then

$$A^{[6]} = \begin{pmatrix} 0 & a^4 & a^2 & 0 \\ a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a & 0 & a^4 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 \end{pmatrix}$$

The last column corresponds to
$2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 2, 1, 2, 2$ and the locator is
$z_1 + a^6x_1^6x_2^2 + a^5x_1^3x_2^2 + x_1^2x_2^2 + a^5x_1^6x_2 + a^3x_2 + a^6x_1^5 + a^3x_1^3$
while the table is

$$A^{[7]} = \begin{pmatrix} 0 & a^5 & a^6 & 0 \\ a^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a^3 & 0 & a^5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a^3 & 0 & 0 \end{pmatrix}$$

The general coefficient matrix for type A configuration is

$$A^{[\text{gen}]} = \begin{pmatrix} 0 & A & B & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C & 0 & A & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 \end{pmatrix}$$

We take as M the value of the third coordinate appearing once, getting

$$\begin{aligned} A &= M^6 \\ B &= M^3 \\ C &= M^5 \end{aligned} \tag{B.90}$$

B.3.3 Type B.

Here we have all the data for Type B configurations from 8.5.

Type B:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

Table B.3: Type B configurations in $\mathbb{F}_8$.

The first column is associated to the list
$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 1$
and to the locator polynomial

$z_1 + x_1^6 x_2^3 + a^3 x_1^3 x_2^3 + a^5 x_1 x_2^3 + a^6 x_2^3 + a^3 x_1^6 x_2^2 + a^5 x_1^4 x_2^2 + a^6 x_1^3 x_2^2$
The coefficients table is

$$\begin{pmatrix} 0 & 0 & a^3 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^5 & 0 \\ 0 & 0 & a^6 & a^3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a^5 \\ 0 & 0 & 0 & a^6 \end{pmatrix}$$

The configuration list obtained by the second column is
$2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1$
while the locator polynomial is
$z_1 + x_1^6 x_2^3 + x_1^3 x_2^3 + x_1 x_2^3 + x_2^3 + x_1^6 x_2^2 + x_1^4 x_2^2 + x_1^3 x_2^2$
and the table turns out to be

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

The third column corresponds to
$2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2$
and the associated locator polynomial is:
$z_1 + x_1^6 x_2^3 + a^4 x_1^3 x_2^3 + a^2 x_1 x_2^3 + a x_2^3 + a^4 x_1^6 x_2^2 + a^2 x_1^4 x_2^2 + a x_1^3 x_2^2$
The coefficients' table is

$$\begin{pmatrix} 0 & 0 & a^4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^2 & 0 \\ 0 & 0 & a & a^4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a^2 \\ 0 & 0 & 0 & a \end{pmatrix}$$

The fourth column is associated to
$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 2, 1$ and to
$z_1 + x_1^6 x_2^3 + a x_1^3 x_2^3 + a^4 x_1 x_2^3 + a^2 x_2^3 + a x_1^6 x_2^2 + a^4 x_1^4 x_2^2 + a^2 x_1^3 x_2^2,$

while the coefficients are represented in

$$\begin{pmatrix} 0 & 0 & a & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^4 & 0 \\ 0 & 0 & a^2 & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a^4 \\ 0 & 0 & 0 & a^2 \end{pmatrix}$$

The following column is linked to the configuration list below
$2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 2, 2, 2, 1, 2, 2.$
The locator polynomial is $z_1 + x_1^6 x_2^3 + a^5 x_1^3 x_2^3 + a^6 x_1 x_2^3 + a^3 x_2^3 + a^5 x_1^6 x_2^2 + a^6 x_1^4 x_2^2 + a^3 x_1^3 x_2^2$
and its coefficients are represented in

$$\begin{pmatrix} 0 & 0 & a^5 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^6 & 0 \\ 0 & 0 & a^3 & a^5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a^6 \\ 0 & 0 & 0 & a^3 \end{pmatrix}$$

The second from last column corresponds to the list
$2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 2,$
to the polynomial
$z_1 + x_1^6 x_2^3 + a^2 x_1^3 x_2^3 + a x_1 x_2^3 + a^4 x_2^3 + a^2 x_1^6 x_2^2 + a x_1^4 x_2^2 + a^4 x_1^3 x_2^2$
and to the table

$$\begin{pmatrix} 0 & 0 & a^2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & a^4 & a^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a \\ 0 & 0 & 0 & a^4 \end{pmatrix}$$

Finally, the last column corresponds to
$2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 1, 2, 1, 2,$
to the polynomial

$z_1 + x_1^6 x_2^3 + a^6 x_1^3 x_2^3 + a^3 x_1 x_2^3 + a^5 x_2^3 + a^6 x_1^6 x_2^2 + a^3 x_1^4 x_2^2 + a^5 x_1^3 x_2^2$
and to

$$\begin{pmatrix} 0 & 0 & a^6 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^3 & 0 \\ 0 & 0 & a^5 & a^6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a^3 \\ 0 & 0 & 0 & a^5 \end{pmatrix}$$

We can summarize the coefficients' tables as

$$\begin{pmatrix} 0 & 0 & A & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & C & A \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B \\ 0 & 0 & 0 & C \end{pmatrix}.$$

As done for type A, we can choose as a value M the value occurring once as a third coordinate and we get

$$\begin{aligned} A &= M^3 \\ B &= M^5 \\ C &= M^6 \end{aligned} \tag{B.91}$$

so, we have again the couple of cycles (β, γ).
Let us now focus on the boldface column i.e. the second one.
As one can easily see by the configuration list, the couples (z_1, z_2) we choose are

$(a,0)$	$(a^2,0)$	$(a^4,0)$
(a,a^2)	(a^2,a^4)	(a^4,a)
(a,a^6)	(a^2,a^5)	(a^4,a^3)
$(a,1)$	$(a^2,1)$	$(a^4,1)$
$(a^3,0)$	$(a^6,0)$	$(a^5,0)$
(a^3,a)	(a^6,a^2)	(a^5,a^4)
(a^3,a^2)	(a^6,a^4)	(a^5,a)
(a^3,a^5)	(a^6,a^3)	(a^5,a^6)
$(a^3,1)$	$(a^6,1)$	$(a^5,1)$
$(1,0)$		

As in type A configurations, only the choices made for a, a^3, 1 are independent, while the other ones come by applying Frobenius mapping.
For example

$$(a,a^2),(a^2,a^4),(a^4,a)$$

is such that
$(a^2,a^4)=(\sigma(a),\sigma(a^2))$
$(a^4,a)=(\sigma(a^2),\sigma(a^4))=(\sigma(\sigma(a)),\sigma(\sigma(a^2)))$.
The other type B configurations come from the optimal Frobenius configuration by cyclic permutations.

B.3.4 Type C.

Here we have all the data for Type C configurations from 8.5.

Type C:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
4	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
4	a^5	$\mathbf{a^4}$	a^3	a^2	a	1	a^6
5	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
5	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
5	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a

Table B.4: Type C configurations in $\mathbb{F}_8$.

The configuration list associated to the first column is:
$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 1, 1, 2, 2$
and the locator polynomial is
$z_1 + x_1^6x_2^3 + a^3x_1^6x_2^2 + a^5x_1^4x_2^2 + a^6x_1^6x_2 + a^3x_1^2x_2 + a^5x_1^3 + a^6x_1^2$.
The table representing its coefficients is

$$\begin{pmatrix} 0 & a^6 & a^3 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^5 & 0 \\ a^5 & 0 & 0 & 0 \\ a^6 & a^3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The second column gives a configuration list
$2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1$
and, as locator,
$z_1 + x_1^6x_2^3 + x_1^6x_2^2 + x_1^4x_2^2 + x_1^6x_2 + x_1^2x_2 + x_1^3 + x_1^2$,

whose corresponding table is

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

The third column gives

$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2$, the locator

$z_1 + x_1^6x_2^3 + a^4x_1^6x_2^2 + a^2x_1^4x_2^2 + ax_1^6x_2 + a^4x_1^2x_2 + a^2x_1^3 + ax_1^2$

and the table

$$\begin{pmatrix} 0 & a & a^4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^2 & 0 \\ a^2 & 0 & 0 & 0 \\ a & a^4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

For the fourth column we have

$2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2$

and the locator polynomial

$z_1 + x_1^6x_2^3 + ax_1^6x_2^2 + a^4x_1^4x_2^2 + a^2x_1^6x_2 + ax_1^2x_2 + a^4x_1^3 + a^2x_1^2$.

The coefficients' table is

$$\begin{pmatrix} 0 & a^2 & a & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^4 & 0 \\ a^4 & 0 & 0 & 0 \\ a^2 & a & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The third to last column gives the configuration list

$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2$.

The associated locator polynoimal is

$z_1 + x_1^6x_2^3 + a^5x_1^6x_2^2 + a^6x_1^4x_2^2 + a^3x_1^6x_2 + a^5x_1^2x_2 + a^6x_1^3 + a^3x_1^2$

and the corresponding table is

$$\begin{pmatrix} 0 & a^3 & a^5 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^6 & 0 \\ a^6 & 0 & 0 & 0 \\ a^3 & a^5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The second to last column gives rise to the list below
$2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1$ and to the following polynomial
$z_1 + x_1^6x_2^3 + a^2x_1^6x_2^2 + ax_1^4x_2^2 + a^4x_1^6x_2 + a^2x_1^2x_2 + ax_1^3 + a^2x_1^2$,
whose coefficients are grouped as

$$\begin{pmatrix} 0 & a^4 & a^2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 \\ a & 0 & 0 & 0 \\ a^4 & a^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Finally, the last column is associated to
$2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 1, 2, 1$
The corresponding locator polynomial is
$z_1 + x_1^6x_2^3 + a^6x_1^6x_2^2 + a^3x_1^4x_2^2 + a^5x_1^6x_2 + a^6x_1^2x_2 + a^3x_1^3 + a^5x_1^2$
and its table is

$$\begin{pmatrix} 0 & a^5 & a^6 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & a^3 & 0 \\ a^3 & 0 & 0 & 0 \\ a^5 & a^6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

We summarize the coefficients' tables as

$$\begin{pmatrix} 0 & A & B & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 \\ C & 0 & 0 & 0 \\ A & B & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Taking as M the value for the third coordinate appearing once in the configuration, we get

$$\begin{aligned} A &= M^6 \\ B &= M^3 \\ C &= M^5 \end{aligned} \tag{B.92}$$

so we have again the couple (β, γ) of $\mathbb{F}_8$ cycles.
The second column, hightlighted as usual in boldface font, is the optimal Frobenius configurations, from which the others arise by cyclic permutations.
We display here the table of the couples (z_1, z_2) in order to lay great stress on the application of Frobenius mapping:

$(a, 0)$	$(a^2, 0)$	$(a^4, 0)$
(a, a^3)	(a^2, a^6)	(a^4, a^5)
(a, a^4)	(a^2, a)	(a^4, a^2)
$(a, 1)$	$(a^2, 1)$	$(a^4, 1)$
$(a^3, 0)$	$(a^6, 0)$	$(a^5, 0)$
(a^3, a^2)	(a^6, a^4)	(a^5, a)
(a^3, a^4)	(a^6, a)	(a^5, a^3)
(a^3, a^6)	(a^6, a^5)	(a^5, a^3)
$(a^3, 1)$	$(a^6, 1)$	$(a^5, 1)$
$(1, 0)$		

B.3.5 Type D.

Here we have all the data for Type D configurations from 8.5. In the Frobenius configuration of type D, the the numbers for a and a^3 in tableB.5 below are exchanged with respect to type A,B,C.

Type D:

Number of points	Third coordinates						
1	a	$\mathbf{1}$	a^6	a^5	a^4	a^3	a^2
4	a^4	$\mathbf{a^3}$	a^2	a	1	a^6	a^5
4	a^6	$\mathbf{a^5}$	a^4	a^3	a^2	a	1
4	1	$\mathbf{a^6}$	a^5	a^4	a^3	a^2	a
5	a^2	$\mathbf{a}$	1	a^6	a^5	a^4	a^3
5	a^3	$\mathbf{a^2}$	a	1	a^6	a^5	a^4
5	a^5	$\mathbf{a^4}$	a^3	a^2	1	1	a^6

Table B.5: Type D configurations in $\mathbb{F}_8$.

The first column of the table above, corresponds to the configuration list
$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1$
The locator polynomial is
$z_1 + ax_1^5x_2^3 + a^2x_1^4x_2^3 + a^4x_1^2x_2^3 + a^4 * x_1^5x_2^2 + x_1^2x_2^2 + ax_1x_2^2 + a^2x_2^2$
and the table of its coefficients is

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a^4 & a \\ 0 & 0 & 0 & a^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a^4 \\ 0 & 0 & a & 0 \\ 0 & 0 & a^2 & 0 \end{pmatrix}$$

The second column gives rise to the following configuration list:
$2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1$ while the locator polynomial is
$z_1 + x_1^5x_2^3 + x_1^4x_2^3 + x_1^2x_2^3 + x_1^5x_2^2 + x_1^2x_2^2 + x_1x_2^2 + x_2^2$

and the coefficients' table is

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

For the third column we get

$2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2,$

the locator

$z_1 + a^6x_1^5x_2^3 + a^5x_1^4x_2^3 + a^3x_1^2x_2^3 + a^3x_1^5x_2^2 + x_1^2x_2^2 + a^6x_1x_2^2 + a^5x_2^2$

and the table

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a^3 & a^6 \\ 0 & 0 & 0 & a^5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a^3 \\ 0 & 0 & a^6 & 0 \\ 0 & 0 & a^5 & 0 \end{pmatrix}$$

The fourth column corresponds to

$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 2, 2, 2, 1$ The locator is

$z_1 + a^5x_1^5x_2^3 + a^3x_1^4x_2^3 + a^6x_1^2x_2^3 + a^6x_1^5x_2^2 + x_1^2x_2^2 + a^5x_1x_2^2 + a^3x_2^2$

and the associated coefficients' list is

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a^6 & a^5 \\ 0 & 0 & 0 & a^3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a^6 \\ 0 & 0 & a^5 & 0 \\ 0 & 0 & a^3 & 0 \end{pmatrix}$$

For the fifth column

$2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 1, 1, 2$

is the configuration list, while the locator polynomial is

$z_1 + a^4x_1^5x_2^3 + ax_1^4x_2^3 + a^2x_1^2x_2^3 + a^2x_1^5x_2^2 + x_1^2x_2^2 + a^4x_1x_2^2 + ax_2^2$

and the associated table is

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a^2 & a^4 \\ 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a^2 \\ 0 & 0 & a^4 & 0 \\ 0 & 0 & a & 0 \end{pmatrix}$$

The sixth column is associated to
$2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2$,
to the locator
$z_1 + a^3x_1^5x_2^3 + a^6x_1^4x_2^3 + a^5x_1^2x_2^3 + a^5x_1^5x_2^2 + x_1^2x_2^2 + a^3x_1x_2^2 + a^6x_2^2$
and to the table

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a^5 & a^3 \\ 0 & 0 & 0 & a^6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a^5 \\ 0 & 0 & a^3 & 0 \\ 0 & 0 & a^6 & 0 \end{pmatrix}$$

Finally, the last column gives rise to
$2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1, 1$
to
$z_1 + a^2x_1^5x_2^3 + a^4x_1^4x_2^3 + ax_1^2x_2^3 + ax_1^5x_2^2 + x_1^2x_2^2 + a^2x_1x_2^2 + a^4x_2^2$
and to

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & a & a^2 \\ 0 & 0 & 0 & a^4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & a^2 & 0 \\ 0 & 0 & a^4 & 0 \end{pmatrix}$$

We summarize the tables of coefficients for type D configurations as

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & A \\ 0 & 0 & B & 0 \\ 0 & 0 & C & 0 \end{pmatrix}$$

Taking as M the value appearing once in each type D configurationwe get the formulas

$$\begin{aligned} A &= M^4 \\ B &= M^1 \\ C &= M^2 \end{aligned} \tag{B.93}$$

so we have the couple (α, γ) of $\mathbb{F}_8$ cycles.
The second column represents theunique optimal type D Frobenius configuration, as shown in the table

$(a, 0)$	$(a^2, 0)$	$(a^4, 0)$
(a, a^2)	(a^2, a^4)	(a^4, a)
(a, a^3)	(a^2, a^6)	(a^4, a^5)
(a, a^6)	(a^2, a^5)	(a^4, a^3)
$(a, 1)$	$(a^2, 1)$	$(a^4, 1)$
$(a^3, 0)$	$(a^6, 0)$	$(a^5, 0)$
(a^3, a^2)	(a^6, a^4)	(a^5, a)
(a^3, a^5)	(a^6, a^3)	(a^5, a^6)
$(a^3, 1)$	$(a^6, 1)$	$(a^5, 1)$
$(1, 0)$		

Each other type D configuration arises from the optimal Frobenius configuration via cyclic permutations.