Algorithms for Symbolic Computation and their Applications

- Standard Bases over Rings and Rank Tests in Statistics -

Oliver Wienand

Vom Fachbereich Mathematik der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.) genehmigte Dissertation

1. Gutachter: Prof. Dr. Gert-Martin Greuel
2. Gutachter: Prof. Dr. Martin Kreuzer

Vollzug der Promotion: 01.07.2011

D 386

Introduction

The thesis covers two independent topics, namely standard basis over rings and convex rank tests.

The theory of Gröbner bases in polynomial rings over rings is in principle well known. Buchberger's algorithm can be effectively extended to base rings where linear equations are solvable. In this thesis we prove that the modification by Mora, Greuel, Pfister and Schönemann, the MGPS-algorithm for computing standard bases with respect to arbitrary monomial orderings can be extended to such base rings as well. Moreover, we take special care to principal ideal rings, allowing zero divisors, for which we design modified algorithms to compute normal forms, Gröbner bases and syzygies, which are new and much faster than the general algorithm. These algorithms were motivated by current limitations in formal verification of microelectronic System-on-Chip designs. We show that our methods from computational algebra are able to overcome these limitations in important classes of applications coming from industrial challenges. In industrial microchip-design current standard methods are not able to prove correctness of certain designs concerning arithmetic functions, e. g. multiplication, accumulated multiplication, and so on. Our polynomial normal form computations are however able to verify examples in these areas where all specialized SAT/SMT-tools fail.

The second part is based on research done during a one-year stay at the University of California – Berkeley. Mainly in collaboration with Jason Morton, Bernd Sturmfels and Anne Shiu we devised a general method to describe and compute a certain class of rank tests motivated by statistics. The class of rank tests may be loosely described as being based on computing the number of linear extensions to given partial orders. In order to apply these tests to actual data we developed two algorithms which both compute the required objects faster than any other algorithm known to us. We used our implementation to apply the methodology to gene expression data created at the Stowers Institute for Medical Research. The dataset is concerned with the development of the vertebra in embryos. Our ranking proved valuable to the biologists.

The algorithms for standard bases over rings $\mathbb{Z}, \mathbb{Z}/m (m \in \mathbb{Z})$ and especially $\mathbb{Z}/2^n$ have been implemented in Singular [23]. The rank tests are also implemented (see [55]).

Part One - Standard bases over rings

The ability to compute standard bases is the most important ingredient in many computations in commutative algebra and algebraic geometry. Furthermore, many problems in different fields, such as system theory, optimization, robotics or formal verification may be phrased in a way which is suitable for standard basis computations. Namely the

problem is expressed as a system of polynomial equations and problems thereof.

Gröbner basis (standard basis with global ordering) and an algorithm for its computation (Buchberger's algorithm) were introduced originally by Buchberger [16] to answer several questions regarding polynomial system, such as

1. is the system solvable and
2. the number of solutions, if finite?

A problem coming from formal verification, and posed by the research group of Prof. W. Kunz [89, 87] at the University of Kaiserslautern, was the motivation to take a detailed look at standard bases where the coefficients of the polynomials are from $\mathbb{Z}/2^n$. Since the coefficients may multiply to zero the degree of a polynomial may drop if multiplied with a scalar. Further many elements in $\mathbb{Z}/2^n$ do not have an inverse element and hence one can not norm polynomials to have the leading coefficient equal to one.

The theoretical existence of standard bases is an easy consequence of the noetherian property of the coefficient ring and is well-known. Also, general algorithms to compute Gröbner basis (i. e. standard basis in the special case of global ordering) are described by Adams and Loustaunau [1]. In this book also the special case of principal ideal domains (i. e. without zero divisors) is treated.

We will at first develop the general theory including a normal form algorithm for modules over polynomial rings over arbitrary noetherian rings and free modules thereof. On this basis we give a constructive proof for the analogon to the Buchberger criterion and syzygy theorem which allows the implementation of a general MGPS-algorithm for arbitrary monomial orderings over coefficient rings in which linear equations are solvable. This is the main result in Chapter 1 and shows the algorithmic feasibility of standard bases computations over arbitrary noetherian rings, as long as basic operations in these rings are algorithmically tractable.

Since the main motivation for our research are computations in $\mathbb{Z}/2^n$ we restrict ourself in Chapter 3 to principal ideal rings and give a complete algorithm allowing practical computations in general principal ideal rings. Here several simplifications are possible and the existence of strong standard bases (see Theorem 2.2.23) allows fast normal form computations. Our algorithm directly computes strong standard basis in the following way. We enlarge the set of elements to be considered in order to apply Buchberger's criterion with special gcd-polynomials. We demonstrate the performance of the algorithm in Chapter 3 in comparison to Magma, the only other computer algebra system we found capable of computing Gröbner bases in $\mathbb{Z}/m$. For local orderings we are not aware of any system which can compute with such orderings and to which we could compare our implementation. In this chapter we give also a brief overview of the SINGULAR architecture and describe the necessary changes to implement our algorithm.

In domains with infinitely many elements polynomials and polynomial functions are in one-to-one relation. But in rings R with finite characteristic this is clearly not the case, as there are infinitely many polynomials but only finitely many functions. Hence the ideal I_0 of all vanishing polynomials, i. e. $I_0 = \{f \in R[\mathbf{x}] \mid f(\mathbf{x}) = 0 \text{ for all } \mathbf{x} = (x_1, \ldots, x_k)\}$, is of interest. Furthermore, if we want to apply algebraic methods we need to be able

to efficiently compute normal forms of polynomials with respect to this ideal. In other words we need a Gröbner basis of I_0. In chapter four we construct an explicit minimal strong Gröbner basis of the ideal of vanishing polynomials in the polynomial ring over $\mathbb{Z}/m$ for $m \geq 2$. The proof is done in a purely combinatorial way. A remarkable fact is that the constructed Gröbner basis is independent of the monomial ordering and that the set of leading terms of the constructed Gröbner basis is unique, up to multiplication by units.

Our primary application is the support of existing property checking techniques for microelectronics in the presence of large data path parts, i. e. circuits used for addition or multiplication or combinations thereof.

Property checking is an important tool used in modern circuit design flows to avoid the introduction of bugs into the circuit design in every stage of the design process. It verifies the functional specification of the initial register transfer level (RTL) description of a circuit design. The initial specification of the design that is often given as a more or less informal human readable document is formalized by a set of properties. A systematic methodology ensures that the complete intended behavior of the circuit is covered by the resulting property suite. However, each property describes the required circuit behavior in a well defined scenario. This allows for an early evaluation for parts of the design as soon as they are completed.

State of the art tools in property checking are using highly evolved SAT solvers to proof the equivalence of the property and the design (i. e. equivalence of the specification and the designed circuits). SAT solvers are dependant on the number of possible valid states of the circuit. Arithmetical functions have inherently many valid states as for each set of input values a valid result exists. For example a circuit multiplying two 64 bit number has at least 2^{128} many different valid states, even if the result is truncated to less than 128 bits. Hence current property checking tools are restricted to areas were not "too much" arithmetic is present or matters.

We developed an algorithm based on our research in standard bases over rings to verify properties which are mainly based on arithmetics. The algorithm is described in chapter five and will also provide a counterexample if the verification fails. We were able to prove all arithmetic properties of a commercial chip design (Infineon Tricore 2) with our implementation which was not verifiable with standard industrial tools.

Part Two - Distributive lattices and rank tests

The concrete motive of our research shown in this part was a problem in biology. The microarray technique enabled researchers to measure the expressions of all known genes at several times in a biological process. Since the number of genes is very high (>20,000) a decision which genes have to be considered for further research has to be taken. Further research consists for example in producing specialized markers for certain genes. The steps for a detailed analysis require a lot of effort. Hence a ranking of the genes according to the likelihood that they are involved in the process under investigation is required. Rank tests give rise to such a ranking in a non-parametric way.

Mary-Lee Dequéant and Olivier Pourquié from the Stowers Institute for Medical Research were researching processes and pathways regarding the development of vertebra within embryos. Therefore time series of 15 to 20 data points, consisting of roughly 20,000 to 30,000 mRNA expression levels, were generated at the Stowers Institute for Medical Research [25]. Our algorithms were able to analyze and rank the different mRNA entries and proved to be helpful for further analysis (see [24]).

Other fields of applications are in general all comparisons between objects using multidimensional data. For example, a comparison of countries regarding several indicators is not unique and depends on the significance or weights which are assigned to the different indicators. Rank tests can help to understand this influence better and give a non-parametric ranking without the need to introduce some way of comparing different indicators. More details on further applications are given in the introduction to the second part.

The ranking algorithms devised are based on computing the number of linear extension for one or many partial orders. The problem is well known in computer science to be #P-complete [10] and optimal algorithms for listing the linear extensions have been devised (see [67]). However, it is possible to compute the number of linear extensions required for many applications by order of magnitudes faster when using a completion based method on distributive lattices developed in this thesis. The corresponding algorithm is able to compute the previously unknown number $|L(2^{[5]})| = 14,807,804,035,657,359,360$ (see [82]) in less than a second and also $|L(2^{[6]})|$, given enough memory and time.

The research in convex rank tests, a special subclass of tests which can be reduced to counting the number of linear extensions to a given or to many partially ordered sets, has lead to answers to three questions posed by Studený, Postnikov, Reiner, and Williams presented in chapter seven.

Organization of the material

Part 1 covers the theory, algorithms, implementation and applications of standard basis for arbitrary monomial orderings where the coefficients form a ring, possible with zero divisors.

In Chapter 1 we introduce the main objects which we shall study in this part, i.e. standard bases, and give a general algorithm for computation which only requires the coefficient ring to be noetherian and the ability to solve linear equations.

Chapter 2 is devoted to the standard basis computation where the coefficients form a noetherian principal ring. In this case the task to solve general linear equations can be reduced to divisibility tests. Further improvements due to the known structure of the syzygy module are possible and are given in form of criteria to omit critical elements.

In Chapter 3 we will describe combinatorial the Gröbner basis of all polynomials evaluating constantly to zero. These polynomials are of interest if we model fixed bit-width arithmetic. This arithmetic can naturally be modeled in the ring $Z/2^n$ with n typically near 32, 64 or 128. In the model we are only interested in the polynomial function but not the polynomial itself. Hence we need methods to factor out the polynomials which define the constant zero function.

Chapter 4 finally concerns algorithms for property checking, based on computer algebra. These are the main applications of the presented algorithms and were also the motivation for the research in standard bases with coefficients in rings.

Part 2 covers research in convex rank tests and applications to computational biology done during a visiting scholarship at the University of California – Berkeley together with Anne Shiu, Bernd Sturmfels, Jason Morton and Lior Pachter.

In Chapter 6 we introduce convex rank tests and describe how they may be used in computational biology to assist further research.

In Chapter 7 we give counterexamples to problems posed by Studený and Postnikov/Reiner/Williams. The counterexamples are a result of research and investigations done while developing convex rank tests. Especially created algorithms have been used to enumerate certain structures in the search for counterexamples.

The last chapter presents the algorithms used for the actual computations in the previous chapters and puts the required computation in the broader context of computer science and computability theory.

Acknowledgements

First of all, I should like to express my gratitude to Prof. Dr. Gert-Martin Greuel and Prof. Dr. Bernd Sturmfels. I am very glad that Gert-Martin has introduced me to this interesting subject, and I should like to thank him for his continuous support and encouragement. Also I should like to thank Markus Wedler for many discussions, explanations and examples regarding formal verification and Frank Seelisch for his support creating this thesis. The main results of Chapter I.4 and I.5 have been published in [37] and [89] during the writing up of this thesis and I like to thank my coauthors of these articles.

Furthermore I owe very much to many valuable discussions with Anne Shiu, Bernd Sturmfels, Jason Morton and Lior Pachter during my stay at University of Berkeley in 2006 which led to all publications of Part 2, cf. [26, 40, 55, 54]. I should also like to thank the Wipprecht foundations which supported my visit at the University of Berkeley.

Many thanks also to my colleagues in the algebraic geometry working group and the Department of Mathematics in Kaiserslautern for providing such a great working atmosphere and always having time to discuss questions.

Zum Schluss möchte ich gerne meiner Frau, meiner Familie und meinen Freunden für ihre Unterstützung herzlich danken.

Contents

Part I.

Standard bases over rings

Introduction to Part I

Many algorithms in computer algebra depend on the ability to compute standard bases. In this part we will first introduce the theory of standard bases for polynomial rings over arbitrary noetherian rings (coefficient rings) and free modules thereof. We will also show that standard bases for arbitrary orderings may be computed, given computations in the coefficient ring are feasible.

In Chapter 2 we concentrate on noetherian principal coefficient rings. For these rings we can give a combinatorial generating set for the syzygies of a given list of elements. This allows further simplifications in the algorithms and leads to an implementation useful for applications.

The next chapter is devoted to the actual implementation in the computer algebra system SINGULAR. We give a brief overview of the architecture of the system related to standard basis computation and describe were changes and extensions are required to compute standard basis with coefficients from noetherian principal ideal rings. Furthermore, we compare the runtime of the implementation of our algorithms to third-party computer algebra systems to demonstrate the capabilities of the presented methods.

In Chapter 4 we will combinatorially describe the Gröbner basis of all polynomials evaluating constantly to zero. These polynomials are of interest if we model fixed bit-width arithmetic. This arithmetic can naturally be modeled in the ring $Z/2^n$ with n typically near 32, 64 or 128. In such a model we are only interested in the polynomial function but not the polynomial itself. Hence we need methods to factor out the polynomials which define the constant zero function.

The last chapter finally concerns the main application, namely property checking, which was one of the reasons for the research presented in the previous chapters.

Property checking has become well-established in modern design flows for Systems-on-Chips (SoCs). Its main application domain is ensuring the correctness of the individual SoC blocks. Nowadays, formal property checking can handle almost all types of modules that can be found in today's SoCs. Nonetheless, a few "pathological" cases remain that sometimes limit the application of property checking in industrial practice. In particular, data paths are often a challenge for formal techniques, especially, if not only the correctness of the control flow but also correctness of the data is to be proved.

For designs containing data paths it is feasible to utilize reasoning and algorithms from computer algebra to actually prove correctness of the designs under scrutiny as described in the last chapter of this part.

Chapter 1.

Theory

In this chapter all necessary definitions and notations will be introduced to prove Buchberger's criterion and the syzygy theorem (see Theorem 1.4.4) for arbitrary monomial orderings and noetherian coefficient rings. The main benefit of the presented proof is its constructive nature, which allows us to compute standard bases in the nearly arbitrary rings. Besides being noetherian the only condition is the ability to solve linear equations in the coefficients (see Definition 1.3.14).

This extends the results by Greuel and Pfister [35] for fields based on the MGPS-algorithm and by Adams and Loustaunau [1] for rings with global orderings based on Buchberger's algorithm and gives rise to a general algorithm to compute standard basis given that ring arithmetics and linear equations are implemented.

1.1. Monomial orderings and associated rings

Throughout this thesis a ring means an arbitrary commutative noetherian ring with one and is usually denoted by R. We will introduce polynomial rings, monomial orderings and further basic notations in the following section.

1.1.1. Rings and monomial orderings

Definition 1.1.1. *Let R be a set with two distinguished elements $0, 1 \in R$ and two operations $+$ and $\cdot$, such that $(R, +, 0)$ is a commutative group, $(R\backslash\{0\}, \cdot, 1)$ is a commutative semi-group with one 1 and multiplication $\cdot$ is distributive above addition $+$. We call $(R, +, \cdot)$ a **ring**. We write R instead of $(R, +, \cdot)$ where unambiguous.*

*An element $c \in R$ is called a **unit** if there exists an element $u \in R$ with $u \cdot c = 1$. The set of units $\mathrm{E}(R)$ is a group with respect to multiplication $\cdot$ and does not contain 0.*

*An element $c \in R$ is called **zero divisor** if there exists an element $z \in R\backslash\{0\}$ such that $z \cdot c = 0$, otherwise **non-zero divisor**.*

*A ring R is called a **domain** if it has no zero divisors except 0.*

For subsets $M, N \subset R$, we write $M + N := \{m + n \mid m \in M, n \in N\}$ and likewise $M \cdot N := \{m \cdot n \mid m \in M, n \in N\}$ for the set of all possible sums or products of elements from M with elements from N.

Definition 1.1.2. *A subset I of a ring R is called an **ideal**, if $I + I \subset I$ and $R \cdot I \subset I$. R is a **principal ideal ring** (PIR) if every ideal in R can be generated by one element. A **principal ideal domain** (PID) is a domain which is also a principal ideal ring.*

An example for a principal ideal ring which is not a domain is the ring $\mathbb{Z}/\langle m\rangle$, where $m \in \mathbb{Z}$ is not a prime number.

Definition 1.1.3. *A ring C is called **noetherian** if every ascending chain of ideals becomes stationary, i. e. for ideals $I_n, n \in \mathbb{N}$ defining an ascending chain*

$$I_1 \subset I_2 \subset I_3 \subset \dots$$

there exists an $n \in \mathbb{N}$ such that the chain is stationary $I_n = I_{n+1} = I_{n+2} = \dots$ beginning with n.

Remark 1.1.4. *The set of nonunits $\mathrm{NE}(R)$, the set of zero divisors $\mathrm{NT}(R)$ and set of non-zero divisors $\mathrm{NNT}(R) = R \backslash \mathrm{NT}(R)$ are semi-groups with respect to multiplication, i. e. multiplicatively closed. The set of non-zero divisors contains the set of units.*

Proof. Let $a, b \in \mathrm{NE}(R)$ and assume $a \cdot b \in \mathrm{E}(R)$. Then an element u with $u \cdot (a \cdot b) = (u \cdot a) \cdot b = 1$ exists, a contradiction to $b \in \mathrm{NE}(R)$.

Now consider $a, b \in \mathrm{NT}(R)$, i. e. there are z_a, z_b such that $z_a \cdot a = 0 = z_b \cdot b$. Since R is commutative $(z_a \cdot z_b) \cdot (a \cdot b) = 0$ and hence $a \cdot b \in \mathrm{NT}(R)$.

Let $a, b \in \mathrm{NNT}(R)$ and assume $a \cdot b \in \mathrm{NT}(R)$. Then an element z with $z \cdot (a \cdot b) = (z \cdot a) \cdot b = 0$ exists, a contradiction to $b \in \mathrm{NNT}(R)$. Finally consider $a \in \mathrm{E}(R)$ and assume $a \in \mathrm{NT}(R)$, i. e. there exist elements u with $a \cdot u = 1$ and $z \neq 0$ with $z \cdot a = 0$. But since $z = z \cdot 1 = z \cdot a \cdot u = (z \cdot a) \cdot u = 0$ a contradiction arises. □

Definition 1.1.5. *The **annihilator** of an element $a \in R$ is the set of all elements annihilating a, i. e.* $\mathrm{Ann}(a) := \{c \in R \mid a \cdot c = 0\}$.

Lemma 1.1.6. *The annihilator* $\mathrm{Ann}(a)$ *is an ideal for each $a \in R$.*

Proof. Let x, y be elements of the annihilator $\mathrm{Ann}(a)$ of a, i. e. $a \cdot x = 0 = a \cdot y$. Hence $a \cdot (x+y) = 0$ and therefore $x+y \in \mathrm{Ann}(a)$. Further let $r \in R$. We obtain $a \cdot (r \cdot x) = r \cdot (a \cdot x) = 0$ and hence $r \cdot x \in \mathrm{Ann}(a)$. □

Remark 1.1.7. *Consider the ring $R = (\mathbb{Z}/\langle 12\rangle) \times \mathbb{Z}$ with component-wise addition and multiplication. Then we have*

$$\begin{aligned}
\mathrm{E}(R) &= \{1,5,7,11\} \times \{-1,1\},\\
\mathrm{NE}(R) &= \{0,2,3,4,6,8,9,10\} \times \mathbb{Z} \ \cup \ \mathbb{Z}/\langle 12\rangle \times \mathbb{Z} \backslash \{-1,1\},\\
\mathrm{NT}(R) &= \{0,2,3,4,6,8,9,10\} \times \mathbb{Z} \ \cup \ \mathbb{Z}/\langle 12\rangle \times \{0\},\\
\mathrm{NNT}(R) &= \{1,5,7,11\} \times \mathbb{Z} \backslash \{0\}
\end{aligned}$$

and, for example, $\mathrm{Ann}((3,1)) = \{(0,0),(4,0),(8,0)\}$ *or* $\mathrm{Ann}((4,0)) = \{0,3,6,9\} \times \mathbb{Z}$.

Definition 1.1.8. *A **polynomial ring** $R[\mathbf{x}]$ with n variables $\mathbf{x} = x_1, x_2, \ldots, x_n$ over a ring R consists of all finite sums $\sum a_\alpha \mathbf{x}^\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n, \mathbf{x}^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $a_\alpha \in R$ for all α.*

Addition is defined component-wise, that is, we add $a_\alpha \mathbf{x}^\alpha$ and $b_\beta \mathbf{x}^\beta$ whenever $\alpha = \beta$ and write $(a_\alpha + b_\beta)\mathbf{x}^\alpha$. Multiplication of monomials works by addition of exponents, multiplication of polynomials is then defined by applying the distributive law.

*The **support** of a polynomial $f = \sum a_\alpha \mathbf{x}^\alpha$ is the set of all monomials with non-zero coefficient, i. e.* $\operatorname{supp}(f) := \{x^\alpha \mid a_\alpha \neq 0\}$.

*The **(total) degree** of a polynomial f is defined by*

$$\deg f := \max\{|\alpha| \mid \mathbf{x}^\alpha \in \operatorname{supp}(f)\},\ |\alpha| := \alpha_1 + \cdots + \alpha_n \text{ if } f \neq 0$$

and else $\deg 0 := -\infty$.

Any linear ordering (total ordering) on $\mathbb{N}^n$ will yield a unique way of writing a given polynomial. In the following we require the ordering to respect monomial multiplication.

Definition 1.1.9. *A linear ordering $<$ on the set of monomials in variables $x_1, x_2, \ldots, x_n$,* $\operatorname{Mon}(\mathbf{x}) := \{\mathbf{x}^\alpha \mid \alpha \in \mathbb{N}^n\}$, *is called a **monomial ordering** on* $\operatorname{Mon}(\mathbf{x})$ *or $R[\mathbf{x}]$ if*

$$\mathbf{x}^\alpha < \mathbf{x}^\beta \Longrightarrow \mathbf{x}^\gamma \mathbf{x}^\alpha < \mathbf{x}^\gamma \mathbf{x}^\beta \tag{1.1}$$

for all $\alpha, \beta, \gamma \in \mathbb{N}^n$. If $1 = \mathbf{x}^0 \leq \mathbf{x}^\alpha$ for all $\alpha \in \mathbb{N}^n$, the monomial ordering is called global. If instead $1 = \mathbf{x}^0 \geq \mathbf{x}^\alpha$ for all $\alpha \in \mathbb{N}^n$ we call $<$ local. If neither is the case the ordering is called mixed.

Lemma 1.1.10. *Let $M \subset \mathbb{N}^n$ be any subset and $\leq_{nat}$ the natural partial ordering on $\mathbb{N}^n$, with $(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq_{nat} (\beta_1, \beta_2, \ldots, \beta_n)$ if and only if $\alpha_i \leq \beta_i$ for all i. Then there is a finite set $B \subset M$ satisfying*

$$\forall \alpha \in M\ \exists \beta \in B \text{ such that } \beta \leq_{nat} \alpha.$$

*B is called a **Dickson basis** of M.*

Proof. See [35] Lemma 1.2.6. □

Definition 1.1.11. *An ordering on M is a **well-ordering** if for each subset $A \subset M$ a minimal element $\hat{a} \in A$ exists, i. e. $a \leq \hat{a}$ for all $a \in A$.*

Corollary 1.1.12. *A monomial ordering is global if and only if it is a well-ordering, i. e. every set of monomials has a minimal element.*

Proof. Let us first fix a global monomial ordering $<$ and some set of monomials $M \subset \operatorname{Mon}(\mathbf{x})$. We check that $<$ refines the natural ordering on $\mathbb{N}^n$:

$$\alpha \leq_{nat} \beta \Rightarrow\ 0 \leq_{nat} \beta - \alpha\ \Rightarrow\ 1 = \mathbf{x}^0 \leq \mathbf{x}^{\beta-\alpha}\ \Rightarrow\ \mathbf{x}^\alpha \leq \mathbf{x}^\beta,$$

where the last implication uses the multiplicative property (1.1).
Writing $A := \{\alpha \mid \mathbf{x}^\alpha \in M\}$, Lemma 1.1.10 provides a finite Dickson basis B for A. The

corresponding set of monomials $N := \{\mathbf{x}^\beta \mid \beta \in B\} \subset M$ is totally ordered by $<$. Hence there is a minimal element in N which is also minimal in M.
Now let $<$ be a monomial well-ordering and assume that $<$ is not global. Then there exists some $\mathbf{x}^\alpha < 1$, i. e.

$$1 > \mathbf{x}^\alpha > \mathbf{x}^{2\alpha} > \mathbf{x}^{3\alpha} > \dots .$$

Hence the set $\{\mathbf{x}^{k\alpha} \mid k \in \mathbb{N}\}$ is no bounded from below. This is a contradiction to $<$ being a well-ordering. □

Example 1.1.13. *The most common ordering may be the lexicographical one (phone book ordering). In case of multivariate polynomials it is defined by*

$$\mathbf{x}^\alpha < \mathbf{x}^\beta :\Longleftrightarrow \exists k : (\forall i < k : \alpha_i = \beta_i) \wedge \alpha_k < \beta_k.$$

It is also natural to consider degree compatible orderings, i. e. orderings with the property $\deg \mathbf{x}^\alpha < \deg \mathbf{x}^\beta \implies \mathbf{x}^\alpha < \mathbf{x}^\beta$. *An example of such an ordering is the degree lexicographical ordering defined by*

$$\begin{aligned}\mathbf{x}^\alpha < \mathbf{x}^\beta :\Longleftrightarrow &\deg \mathbf{x}^\alpha < \deg \mathbf{x}^\beta \text{ or} \\ &\deg \mathbf{x}^\alpha = \deg \mathbf{x}^\beta \wedge \exists k : (\forall i < k : \alpha_i = \beta_i) \wedge \alpha_k < \beta_k.\end{aligned}$$

Another important ordering is the degree reverse lexicographical ordering, defined by

$$\begin{aligned}\mathbf{x}^\alpha < \mathbf{x}^\beta :\Longleftrightarrow &\deg \mathbf{x}^\alpha < \deg \mathbf{x}^\beta \text{ or} \\ &\deg \mathbf{x}^\alpha = \deg \mathbf{x}^\beta \wedge \exists k : (\forall i > k : \alpha_i = \beta_i) \wedge \alpha_k > \beta_k.\end{aligned}$$

Lemma 1.1.14. *Let $R[\mathbf{x}]$ be a polynomial ring, and polynomials $f, g \in R[\mathbf{x}]$. Then*

$$\operatorname{supp}(f+g) \subset \operatorname{supp}(f) \cup \operatorname{supp}(g) \tag{1.2}$$
$$\operatorname{supp}(f \cdot g) \subset \operatorname{supp}(f) \cdot \operatorname{supp}(g) \tag{1.3}$$

Moreover, if R is a domain then

$$\operatorname{supp}(f \cdot g) = \operatorname{supp}(f) \cdot \operatorname{supp}(g). \tag{1.4}$$

Proof. Let $f = \sum_{\alpha \in I} a_\alpha \mathbf{x}^\alpha$ and $g = \sum_{\alpha \in I} b_\alpha \mathbf{x}^\alpha$. If the sum of two coefficients a_α, b_α is not equal to zero at least one of the coefficients has to be different from zero. Hence

$$\begin{aligned}\operatorname{supp}(f+g) &= \{\mathbf{x}^\alpha \mid \alpha \in I \text{ and } a_\alpha + b_\alpha \neq 0\} \\ &\subset \{\mathbf{x}^\alpha \mid \alpha \in I \text{ and } a_\alpha \neq 0\} \cup \{\mathbf{x}^\alpha \mid \alpha \in I \text{ and } b_\alpha \neq 0\} \\ &= \operatorname{supp}(f) \cup \operatorname{supp}(g).\end{aligned}$$

Analogously in the case of multiplication

$$\begin{aligned}\operatorname{supp}(f \cdot g) &= \{\mathbf{x}^{\alpha+\beta} = \mathbf{x}^\alpha \cdot \mathbf{x}^\beta \mid \alpha, \beta \in I \text{ and } a_\alpha \cdot b_\beta \neq 0\} \\ &\subset \{\mathbf{x}^\alpha \mid \alpha \in I \text{ and } a_\alpha \neq 0\} \cdot \{\mathbf{x}^\alpha \mid \alpha \in I \text{ and } b_\alpha \neq 0\} \\ &= \operatorname{supp}(f) \cdot \operatorname{supp}(g).\end{aligned}$$

The final statement follows since for a domain the latter inclusion becomes an equality. □

Definition 1.1.15. *Let $R[\mathbf{x}]$ be a polynomial ring with monomial ordering $<$, $f = \sum b_\beta \mathbf{x}^\beta \in R[\mathbf{x}]$ a polynomial, and let $\mathbf{x}^\alpha \in \operatorname{supp}(f)$ denote the unique monomial in $\operatorname{supp}(f)$ satisfying $\mathbf{x}^\beta \leq \mathbf{x}^\alpha$ for all $\beta \in \operatorname{supp}(f)$. Then we define the following symbols:*

$$\begin{aligned}
\operatorname{LM}(f) &:= \mathbf{x}^\alpha && \textit{leading monomial of } f,\\
\operatorname{LE}(f) &:= \alpha && \textit{leading exponent of } f,\\
\operatorname{LC}(f) &:= a_\alpha && \textit{leading coefficient of } f,\\
\operatorname{LT}(f) &:= a_\alpha \cdot \mathbf{x}^\alpha && \textit{leading term of } f,\\
\operatorname{tail}(f) &:= f - \operatorname{LT}(f) && \textit{tail of } f.
\end{aligned}$$

Lemma 1.1.16. *Let $f, g \in R[\mathbf{x}]$. Then $\operatorname{LM}(f+g) \leq \max\{\operatorname{LM}(f), \operatorname{LM}(g)\}$, respectively $\operatorname{LM}(f \cdot g) \leq \operatorname{LM}(f) \cdot \operatorname{LM}(g)$, and equality holds if $\operatorname{LT}(f) + \operatorname{LT}(g) \neq 0$, respectively $\operatorname{LC}(f) \cdot \operatorname{LC}(g) \neq 0$.*

Proof. Lemma 1.1.14 states

$$\begin{aligned}
\operatorname{supp}(f+g) &\subset \operatorname{supp}(f) \cup \operatorname{supp}(g)\\
\operatorname{supp}(f \cdot g) &\subset \operatorname{supp}(f) \cdot \operatorname{supp}(g)
\end{aligned}$$

and hence

$$\begin{aligned}
\operatorname{LM}(f+g) &= \max \operatorname{supp}(f+g)\\
&\leq \max\{\max \operatorname{supp}(f), \max \operatorname{supp}(g)\}\\
&= \max\{\operatorname{LM}(f), \operatorname{LM}(g)\}.\\
\operatorname{LM}(f \cdot g) &= \max \operatorname{supp}(f \cdot g)\\
&\leq \max(\operatorname{supp}(f) \cdot \operatorname{supp}(g))\\
&= \max \operatorname{supp}(f) \cdot \max \operatorname{supp}(g),\\
&\quad \text{since } < \text{ is a monomial ordering and respects multiplication}\\
&= \operatorname{LM}(f) \cdot \operatorname{LM}(g).
\end{aligned}$$

If $\operatorname{LT}(f) + \operatorname{LT}(g) \neq 0$ then $\max\{\operatorname{LM}(f), \operatorname{LM}(g)\} \in \operatorname{supp}(f+g)$ and likewise $\operatorname{LM}(f) \cdot \operatorname{LM}(g) \in \operatorname{supp}(f \cdot g)$ if $\operatorname{LC}(f) \cdot \operatorname{LC}(g) \neq 0$. □

1.1.2. Rings associated to monomial orderings

Now we take a look on localizations and how every monomial ordering defines a localization on $R[\mathbf{x}]$. In contrast to the case where R is a field, we have to take special care due to the presence of zero divisors in the ring R.

Localizations

Definition 1.1.17. *Let R be a ring and S a multiplicative set, i.e. $S \subset R$ is a multiplicatively closed subset with one, i. e. $1 \in S \cdot S \subset S$. The **localization** of R by S is defined by*

$$S^{-1}R := \left\{ \frac{a}{b} \;\middle|\; a \in R \text{ and } b \in S \right\}$$

with the identification $\frac{a_1}{b_1} = \frac{a_2}{b_2}$ if $\exists h \in S \;:\; h \cdot (a_1 b_2 - a_2 b_1) = 0$.

Example 1.1.18. *The field of rational numbers $\mathbb{Q}$ is the localization of the integers $\mathbb{Z}$ by $\mathbb{Z}\backslash\{0\}$. The localization by the set of non-zero divisors of a ring R is also called the **total quotient ring** $\mathrm{Quot}(R)$ of R.*

In presence of zero divisors, localizations are a bit more subtle. For example consider the localization of $\mathbb{Z}/12$ by $S = \{1, 3, 9\}$. The resulting ring is isomorphic to $\mathbb{Z}/4$. The equivalence relation is

$$\begin{aligned}
0 = 4 = 8 &= \frac{0}{3} = \frac{4}{3} = \frac{8}{3} = \frac{0}{9} = \frac{4}{9} = \frac{8}{9} \\
1 = 5 = 9 &= \frac{3}{3} = \frac{7}{3} = \frac{11}{3} = \frac{1}{9} = \frac{5}{9} = \frac{9}{9} \\
2 = 6 = 10 &= \frac{2}{3} = \frac{6}{3} = \frac{10}{3} = \frac{2}{9} = \frac{6}{9} = \frac{10}{9} \\
3 = 7 = 11 &= \frac{1}{3} = \frac{5}{3} = \frac{9}{3} = \frac{3}{9} = \frac{7}{9} = \frac{11}{9}.
\end{aligned}$$

Lemma 1.1.19. *The localization of the ring $\mathbb{Z}/m$ by a set $S \subset \mathbb{Z}/m$ is isomorphic to the ring $\mathbb{Z}/\frac{m}{k}$ with $k = \max\{\gcd(s, m) \mid [s] \in S\}$.*

Here $[s]$ denotes the residue class of $s \in \mathbb{Z}$ and $\gcd(s, m)$ denotes the unique positive greatest common divisor of s and m.

Proof. We will proof that the map

$$\begin{aligned}
\varphi : \mathbb{Z}/\frac{m}{k} &\longrightarrow S^{-1}\mathbb{Z}/m \\
[a]_{m/k} &\longmapsto \frac{[a]_m}{[1]_m} = [a]_m
\end{aligned}$$

is an isomorphism of rings. We have to show that φ is well-defined, injective and surjective. The additive and multiplicative properties follow directly from the definition. Let $[a]_{m/k} = [b]_{m/k}$. We have to show that there exists an $[s]_m \in S$ such that

$$[s]_m([1]_m[a]_m - [1]_m[b]_m) = [0]_m.$$

Let $[s]_m \in S$ be such, that $\gcd(s, m) = k$. Since $\frac{m}{k} \mid a - b$ we know $m \mid s \cdot (a - b)$ and hence $[s \cdot (a - b)]_m = [0]_m$, i. e. $\varphi([a]_{m/k}) = \varphi([b]_{m/k})$.

Now assume $\varphi([a]_{m/k}) = \varphi([b]_{m/k})$ for arbitrary $a, b \in \mathbb{Z}$, i. e. there exists an $[s]_m \in S$ with $[s]_m([1]_m[a]_m - [1]_m[b]_m) = [0]_m$. Hence $m \mid s \cdot (a - b)$. Since S is multiplicatively closed we know $\gcd(s, m) \mid k$ and therefore $\frac{m}{k} \mid a - b$, i. e. $[a]_{m/k} = [b]_{m/k}$.

At last we have to show, that for all $\frac{[a]_m}{[b]_m}$ there exists a $[c]_m = \frac{[a]_m}{[b]_m}$, i. e. $[s]_m([b]_m[c]_m - [a]_m) = [0]_m$ for a $[s]_m \in S$. Choose $[s]_m \in S$ with $\gcd(s, m) = k$. Since k is maximal for elements in S and S is multiplicatively closed we obtain $\gcd(s \cdot b, m) = k$ and hence

$$[s \cdot b]_m = [u_1]_m[k]_m \qquad [s]_m = [u_2]_m[k]_m$$

with $\gcd(u_1, m) = \gcd(u_2, m) = 1$, i. e. units $[u_1]_m, [u_2]_m \in \mathbb{Z}/m$.

Let $[c]_m = [u_1]_m^{-1}[u_2]_m[a]_m$. Then

$$[s]_m([b]_m[c]_m - [a]_m) = [u_1]_m[k]_m[u_1]_m^{-1}[u_2]_m[a]_m - [u_2]_m[k]_m[a]_m = 0.$$

Therefore φ is also surjective and hence an isomorphism of rings. □

The previous lemma shows that localizations of rings $\mathbb{Z}/m$ do not give rise to rings outside of the family $\mathbb{Z}/m$. The above result could also be achieved by the Chinese Remainder Theorem which also proves, that $\mathbb{Z}/m$ is isomorphic to all possible localizations of itself.

Localizations defined by monomial orderings

Lemma 1.1.20. *Let $R[\mathbf{x}]$ be a polynomial ring with monomial ordering $<$. The sets*

$$S_< = \{f \in R[\mathbf{x}] \mid \mathrm{LM}(f) = 1 \wedge \mathrm{LC}(f) \in \mathrm{E}(R)\} \text{ and}$$
$$\hat{S}_< = \{f \in R[\mathbf{x}] \mid \mathrm{LM}(f) = 1 \wedge \mathrm{LC}(f) \in \mathrm{NNT}(R)\}$$

are multiplicatively closed and contain one.

Proof. Definition 1.1.1 and Remark 1.1.4 state $1 \in \mathrm{E}(R), 1 \in \mathrm{NNT}(R)$ and that the sets $\mathrm{E}(R), \mathrm{NNT}(R)$ are multiplicatively closed. Therefore $1 \in S_<, \hat{S}_<$ and we have $\mathrm{LC}(f) \cdot \mathrm{LC}(g) \in \mathrm{E}(R)$ for $f, g \in S_<$, respectively $\mathrm{LC}(f) \cdot \mathrm{LC}(g) \in \mathrm{NNT}(R)$ for $f, g \in \hat{S}_<$.
Further $0 \notin \mathrm{E}(R), 0 \notin \mathrm{NNT}(R)$ and therefore Lemma 1.1.16 yields $\mathrm{LM}(f \cdot g) = \mathrm{LM}(f) \cdot \mathrm{LM}(g) = 1$ for both $f, g \in S_<$ and $f, g \in \hat{S}_<$. Hence $f \cdot g \in S_<$, respectively $f \cdot g \in \hat{S}_<$. □

Definition 1.1.21. *Let $R[\mathbf{x}]$ be a polynomial ring with monomial ordering $<$ and $S_<$ as in Lemma 1.1.20. Let*

$$R[\mathbf{x}]_< := {S_<}^{-1}R[\mathbf{x}] = \left\{ \frac{f}{g} \;\middle|\; f \in R[\mathbf{x}], g \in S_< \right\}$$
$$= \left\{ \frac{f}{g} \;\middle|\; f, g \in R[\mathbf{x}], \mathrm{LM}(g) = 1 \wedge \mathrm{LC}(g) \in \mathrm{E}(R) \right\}$$

*denote the localization of $R[\mathbf{x}]$ by $S_<$, with $\frac{f_1}{g_1} = \frac{f_2}{g_2}$ iff $\exists h : h \cdot (f_1g_2 - f_2g_1) = 0$. The ring $R[\mathbf{x}]_<$ is called the **associated ring** to the monomial ordering $<$.*

Since $S_<$ and $\hat{S}_<$ are both multiplicative, we can use both for the localization of $R[\mathbf{x}]$. We may call ${S_<}^{-1}R[\mathbf{x}]$ the **weak localization** and $\left(\hat{S}_<\right)^{-1} R[\mathbf{x}]$ the **strong localization** of $R[\mathbf{x}]$.

The following lemma shows how the strong localization can be derived from the weak localization.

Lemma 1.1.22. *Let $\hat{R} = \mathrm{Quot}(R)$ be the total quotient ring of R, i. e. the localization of R by $\mathrm{NNT}(R)$. Then $\hat{R}[\mathbf{x}]_< = \left(\hat{S}_<\right)^{-1} R[\mathbf{x}]$ for any monomial ordering. In particular*

$$R[\mathbf{x}]_< = S_<^{-1}R[\mathbf{x}] \subset \left(\hat{S}_<\right)^{-1} R[\mathbf{x}] = \hat{R}[\mathbf{x}]_<.$$

Proof. The inclusion follows directly from the obvious fact $S_< \subset \hat{S}_<$. We show the right-hand side of the equation.
Consider an element $\frac{f}{g} \in \hat{R}[\mathbf{x}]_<$. We write $f = \sum a_\alpha/b_\alpha \cdot \mathbf{x}^\alpha$ and $g = \sum c_\beta/d_\beta \cdot \mathbf{x}^\beta$. Then, by assumption, $\mathrm{LM}(g) = 1, \mathrm{LC}(g) \in \mathrm{E}\left(\hat{R}\right)$. We set $B := \prod b_\alpha, D := \prod d_\beta$, $a'_\alpha := a_\alpha/b_\alpha \cdot B$ and know $B, D \neq 0$. Now let $c'_\beta := c_\beta/d_\beta \cdot D$ and compute

$$\frac{f}{g} = \frac{\sum \frac{a_\alpha}{b_\alpha}\mathbf{x}^\alpha}{\sum \frac{c_\beta}{d_\beta}\mathbf{x}^\beta} = \frac{\frac{1}{B}\sum a'_\alpha \mathbf{x}^\alpha}{\frac{1}{D}\sum c'_\beta \mathbf{x}^\beta} = \frac{\sum D \cdot a'_\alpha \mathbf{x}^\alpha}{\sum B \cdot c'_\beta \mathbf{x}^\beta}$$

Let us denote the last denominator by h. Then it remains to show, that $h \in \hat{S}_<$.
We pick β_0 such that $\mathrm{LT}(g) = c_{\beta_0}/d_{\beta_0} \cdot \mathbf{x}^{\beta_0}$. Then $\mathbf{x}^{\beta_0} = 1$ and $c_{\beta_0}/d_{\beta_0} \in \mathrm{E}\left(\hat{R}\right)$, i. e., there is some $u \in \hat{R}$ such that $1 = u \cdot c_{\beta_0}/d_{\beta_0} = u/d_{\beta_0} \cdot c_{\beta_0}$. Therefore obviously $c_{\beta_0} \in \mathrm{NNT}(R)$. Note also that $B, D \in \mathrm{NNT}(R)$, and hence $\mathrm{LT}(h) = BD\,\mathrm{LT}(g)$. But then $\mathrm{LM}(h) = 1$ and $\mathrm{LC}(h) = c_{\beta_0} B \prod_{\beta \neq \beta_0} d_\beta \in \mathrm{NNT}(R)$, and $h \in \hat{S}_<$.

For the converse, fix $f/g \in \left(\hat{S}_<\right)^{-1} R[\mathbf{x}]$. We know that $\mathrm{LM}(g) = 1$ and $\mathrm{LC}(g) \in \mathrm{NNT}(R)$. Then, by definition of $\hat{R}$, $\mathrm{LC}(g) \in \mathrm{E}\left(\hat{R}\right)$, and hence $f/g \in \hat{R}[\mathbf{x}]_<$. □

Definition 1.1.23. *Let $f \in R[\mathbf{x}]_<$ and $u \in R[\mathbf{x}]$ such that $\mathrm{LT}(u) = 1$ and $u \cdot f \in R[\mathbf{x}]$. We define $\mathrm{LM}(f) := \mathrm{LM}(u \cdot f)$ and further $\mathrm{LT}, \mathrm{LC}, \mathrm{LE}, \mathrm{tail}$ also as those of $u \cdot f$.*

Remark 1.1.24. *An element u as required in the previous definition always exists and the derived expressions are independent of the choice of u. Similarly, if $f \in R[\mathbf{x}]_<$, there is some $u \in R[\mathbf{x}]$, $\mathrm{LT}(u) = 1$ such that $u \cdot f \in R[\mathbf{x}]$ and we can define the leading data in $R[\mathbf{x}]_<$ by the corresponding data in $R[\mathbf{x}]$.*

Proof. We just show the existence of a polynomial u as claimed. Let $\frac{f}{g} \in R[\mathbf{x}]_<$. By definition $\mathrm{LC}(g) \in \mathrm{E}(R)$ and $\mathrm{LM}(g) = 1$. Therefore $u := \mathrm{LC}(g)^{-1} \cdot g$ has the properties

1. $u \in R[\mathbf{x}], \mathrm{LT}(u) = 1$ and
2. $u \cdot \frac{f}{g} = \mathrm{LC}(g)^{-1} \cdot f \in R[\mathbf{x}]$.

□

Definition 1.1.25. *The **ring of formal power series**, that is, the ring of possibly infinite sums of the form* $\sum_{\alpha\in\mathbb{N}^n}^{\infty} a_\alpha \mathbf{x}^\alpha$*, is defined by monomial-wise addition and the generalized Cauchy product. We denote it by* $R[[\mathbf{x}]]$*.*

Now we compare the localization $R[\mathbf{x}]_<$ with the power series ring $R[[\mathbf{x}]]$. Denote by $R[\mathbf{x}]_{\langle\mathbf{x}\rangle}$ the localization of $R[\mathbf{x}]$ with respect to the multiplicative set $\{f \in R[\mathbf{x}] \mid f(0,\ldots,0) \in \mathrm{E}(R)\}$.

Lemma 1.1.26. *Let R be a ring and $R[\mathbf{x}]$ a polynomial ring with monomial ordering $<$. Then*

a) $R[\mathbf{x}] \subset R[\mathbf{x}]_< \subset R[\mathbf{x}]_{\langle\mathbf{x}\rangle} \subset R[[\mathbf{x}]]$.

b) The set of units $\mathrm{E}(R[\mathbf{x}]_<)$ *is given by*

$$\mathrm{E}(R[\mathbf{x}]_<) = \left\{ \frac{f}{g} \;\middle|\; f,g \in R[\mathbf{x}], \mathrm{LM}(f) = \mathrm{LM}(g) = 1, \mathrm{LC}(f), \mathrm{LC}(g) \in \mathrm{E}(R) \right\}.$$

c) $R[\mathbf{x}] = R[\mathbf{x}]_<$ *if and only if* $<$ *is global and* $R[\mathbf{x}]_< = R[\mathbf{x}]_{\langle\mathbf{x}\rangle}$ *if and only if* $<$ *is local.*

d) $R[\mathbf{x}]_<$ *is noetherian iff R is noetherian.*

e) $R[\mathbf{x}]_<$ *is factorial iff R is factorial.*

f) If $R[t]$ is a principal ideal ring, then so is $R[t]_<$. Moreover, $R[\mathbf{x}]_<$ can only be a principal ideal ring when the number of variables $\mathbf{x}$ is one.

In addition to d), recall that R is noetherian iff $R[\mathbf{x}]$ is noetherian. Concerning f) note that assuming that R is a PIR is in general not sufficient for $R[t]$ to be a PIR: Chose $R = K[s]$ a univariate polynomial ring over some field K. Then R is a PIR but $R[t] \cong K[s,t]$ is not since $\langle s,t\rangle$ has no single generator.

Proof. Claims a) - c) are proved as for the field case; see [35] Lemma 1.5.2. (1)-(3). Only $R[\mathbf{x}]_< \subset R[[\mathbf{x}]]$ is not covered there: Consider $f/g \in R[\mathbf{x}]_{\langle\mathbf{x}\rangle}$. Then $g(\mathbf{x}) = u + h(\mathbf{x})$ with some $u \in \mathrm{E}(R)$ and $h \in \langle x_1,\ldots,x_n\rangle$. Then the power series expansion of $1/g$ yields

$$\frac{f}{g} = \frac{f}{u+h(\mathbf{x})} = \frac{u^{-1}f}{1+u^{-1}h(\mathbf{x})} = u^{-1}f \cdot \sum_{k=0}^{\infty} \left(u^{-1}h(\mathbf{x})\right)^k \in R[[\mathbf{x}]].$$

d) [35] Lemma 1.4.8.(2) shows that $R[\mathbf{x}]_<$ is noetherian if R is. Conversely, let $R[\mathbf{x}]_<$ be noetherian and consider a sequence of ideals $I_1 \subset I_2 \subset \ldots \subset R$. Then $I_1 \cdot R[\mathbf{x}]_< \subset I_2 \cdot R[\mathbf{x}]_< \subset \ldots$ must stabilize, i. e., $I_N \cdot R[\mathbf{x}]_< = I_{N+k} \cdot R[\mathbf{x}]_<, k \geq 0$. Hence

$$a \in I_{N+k} \;\Rightarrow\; a \in I_{N+k} \cdot R[\mathbf{x}]_< \cap R = I_N \cdot R[\mathbf{x}]_< \cap R \;\Rightarrow\; a \in I_N \cdot R \cap R = I_N,$$

and $I_1 \subset I_2 \subset \ldots \subset R$ also stabilizes.
e) See [35] Exercise 1.4.9. for the hard implication. Conversely, it is easy to see that factoriality of $R[\mathbf{x}]_<$ implies factoriality of R.
f) $R[t]$ is a PIR and hence noetherian. Thus $R[t]_<$ is noetherian, too. We fix an ideal $I \subset R[t]_<$ which is then finitely generated, $I = \langle f_1/g_1, \ldots, f_k/g_k \rangle$ for some $k < \infty$. We write $g := g_1 \cdots g_k$ and $h_i := g/g_i$. Then $J := \langle h_1 f_1, \ldots, h_k f_k \rangle \subset R[t]$ is generated by a single polynomial h, that is, $J = \langle h \rangle$. By construction, $h_i f_i = a_i h$ for some $a_1, \ldots, a_k \in R[t]$. This yields $f_i/g_i = h_i f_i/g = a_i h/g$. Now let a denote the single generator of $\langle a_1 \ldots, a_k \rangle$, then $I = \langle ah/g \rangle$ implying that $R[t]_<$ is a PIR.
For the second claim assume for a contradiction that x, y are distinct variables in $\mathbf{x}$, but $\langle x, y \rangle = \langle f/g \rangle$ in $R[\mathbf{x}]_<$. Since $g \in \mathrm{E}(R[\mathbf{x}]_<)$, we can assume $g = 1$. We write $x = a/a' \cdot f, y = b/b' \cdot f$ with $a', b' \in S_<$, i. e., $\mathrm{LT}(a'), \mathrm{LT}(b')$ units in R and hence 1 without loss of generality. We obtain the polynomial equation $a'x = af$ implying that $x = \mathrm{LT}(a'x) = \mathrm{LT}(af)$. We may assume that $\mathrm{LT}(a) f \neq 0$ since we can omit all terms s in a with the property $sf = 0$. But then $x = \mathrm{LT}(a) t$ for some term t in f. This shows that $\mathrm{LC}(a) \in \mathrm{E}(R)$. Therefore, $\mathrm{LT}(a)$ does not kill any term of f and hence $x = \mathrm{LM}(a) \mathrm{LM}(f)$. This gives rise to two cases. If $\mathrm{LM}(a) = 1$ we obtain $a \in \mathrm{E}(R[\mathbf{x}]_<)$ and hence the contradiction $\langle x \rangle = \langle f \rangle$. On the other hand, if $\mathrm{LM}(f) = 1$ then $f \in \mathrm{E}(R[\mathbf{x}]_<)$ and $\langle f \rangle = \langle 1 \rangle = R[\mathbf{x}]_<$, again a contradiction. □

Definition 1.1.27. *Let $R[\mathbf{x}]$ be a polynomial ring with monomial ordering $<$ and $A \subset R[\mathbf{x}]$ or $A \subset R[\mathbf{x}]_<$ any subset. We define the ideals*

$$\mathrm{L}(A) = \langle \mathrm{LT}(f) \mid f \in A \rangle_{R[\mathbf{x}]} \qquad \textit{leading ideal of } A,$$
$$\mathrm{LM}(A) = \langle \mathrm{LM}(f) \mid f \in A \rangle_{R[\mathbf{x}]} \qquad \textit{leading monomials ideal of } A.$$

Note that over a coefficient field, $\mathrm{L}(A) = \mathrm{LM}(A)$ always holds. This is in general not true over coefficient rings, e.g. for $A = \{2x\} \subset \mathbb{Z}[x]$ we have $\mathrm{L}(A) = \langle 2x \rangle$ but $\mathrm{LM}(A) = \langle x \rangle$.

1.2. Modules and syzygies

Definition 1.2.1. *Let R be a ring. A commutative group $(M, +)$ together with a scalar multiplication*

$$R \times M \to M, (f, m) \mapsto f \cdot m,$$

such that

$$f_1 \cdot (f_2 \cdot m) = (f_1 \cdot f_2) \cdot m$$
$$f_1 \cdot m + f_2 \cdot m = (f_1 + f_2) \cdot m$$
$$f \cdot m_1 + f \cdot m_2 = f \cdot (m_1 + m_2)$$
$$1_R \cdot m = m$$

*with $f, f_1, f_2 \in R$ and $m, m_1, m_2 \in M$ is called an R-**module** or a module over R. An R-module is called **free** of **rank** k, if it is isomorphic to R^k with component-wise addition*

and scalar multiplication, for some $k \in \mathbb{N}$. A subset $N \subset M$ is called a ***submodule*** *of $(M,+)$ if $(N,+)$ is itself a module.*

Definition 1.2.2. *A module M over R is* ***generated by*** *a subset $G \subset M$, if M is the set of all finite R-linear combinations of elements from G, i. e.*

$$\langle G \rangle_R := \{ \sum_{finite} a_g \cdot g \mid g \in G, a_g \in R \} = M$$

. We write $\mathrm{Gen}(M)_R$ for an arbitrary generating set and $\mathrm{Gen}_{\mathrm{fin}}(M)_R$ for a finite one (if existent).

Definition 1.2.3. *Let R be a ring and M a module over R. A* ***syzygy*** *between k elements $f_1, \ldots, f_k$ of M is a k-tuple $(g_1, \ldots, g_k) \in R^k$ such that*

$$(f_1, \ldots, f_k) \cdot \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix} = \sum_{i=1}^{k} g_i \cdot f_i = 0.$$

Remark 1.2.4. *The set of syzygies of k given elements f_i, $\mathrm{Syz}(f_1, \ldots, f_k)$, forms a submodule of R^k called the* ***syzygy module****, as the addition of two syzygies and the multiplication of a syzygy with an element of R again yield syzygies. The syzygy module is the kernel of the map*

$$\begin{aligned} \varphi : F_1 := \bigoplus_{i=1}^{k} R\, \epsilon_i &\longrightarrow M \\ \epsilon_i &\mapsto f_i, \end{aligned}$$

whereas the image of φ is the R-module $I := \langle f_1, \ldots, f_k \rangle_R$.

We define $\mathrm{Syz}(I) := \mathrm{Syz}(f_1, \ldots, f_k)$, which is well-defined up to isomorphism (see [35], Remark 2.5.2).

Definition 1.2.5. *Let $R[\mathbf{x}]$ be a polynomial ring with monomial ordering $<$. A* ***module (monomial) ordering*** *$<_m$, of the free module $M = \bigoplus_{i=1}^{m} R[\mathbf{x}] \cdot \mathbf{e}_i$ is a total ordering on the set of monomials $\mathrm{Mon}(M) := \{\mathbf{x}^\alpha \cdot \mathbf{e}_i \mid \alpha \in \mathbb{N}^n, i = 1, \ldots, m\}$ compatible with the module structure and the monomial ordering $<$ of $R[\mathbf{x}]$, that is,*

$$\begin{aligned} \mathbf{x}^\alpha \cdot \mathbf{e}_i <_m \mathbf{x}^\beta \cdot \mathbf{e}_j &\Longrightarrow \mathbf{x}^\gamma \cdot \mathbf{x}^\alpha \cdot \mathbf{e}_i <_m \mathbf{x}^\gamma \cdot \mathbf{x}^\beta \cdot \mathbf{e}_j \\ \mathbf{x}^\alpha < \mathbf{x}^\beta &\Longrightarrow \mathbf{x}^\alpha \cdot \mathbf{e}_i <_m \mathbf{x}^\beta \cdot \mathbf{e}_i. \end{aligned}$$

Example 1.2.6. *The two natural orderings on a free module over $R[\mathbf{x}]$ with ordering $<$ are*

components-first

Module components take precedence over the monomial ordering on $R[\mathbf{x}]$, i. e.

$$\begin{aligned} \mathbf{x}^\alpha \cdot \mathbf{e}_i <_m \mathbf{x}^\beta \cdot \mathbf{e}_j \quad :\Longleftrightarrow \quad & i < j \\ & \text{or } (i = j \text{ and } \mathbf{x}^\alpha < \mathbf{x}^\beta), \end{aligned}$$

components-last
the monomial ordering takes precedence over the module components, i. e.

$$\mathbf{x}^\alpha \cdot \mathbf{e}_i <_m \mathbf{x}^\beta \cdot \mathbf{e}_j \iff \mathbf{x}^\alpha < \mathbf{x}^\beta \text{ or } (\mathbf{x}^\alpha = \mathbf{x}^\beta \text{ and } i < j).$$

But there are also more subtle orderings, like the Schreyer ordering defined below.

Definition 1.2.7. *Let $M = \bigoplus_{i=1}^m R[\mathbf{x}] \cdot \mathbf{e}_i$ be a module with ordering $<$ and $f_1, \ldots, f_k$ be elements of M. Further let $F_1 = \bigoplus_{i=1}^k R[\mathbf{x}] \cdot \epsilon_i$ be the free k-module over $R[\mathbf{x}]$. The ordering $<_1$ on F_1 given by*

$$\mathbf{x}^\alpha \epsilon_i <_1 \mathbf{x}^\beta \epsilon_j \iff \begin{array}{l} \mathrm{LM}(\mathbf{x}^\alpha f_i) < \mathrm{LM}(\mathbf{x}^\beta f_j) \text{ or} \\ \mathrm{LM}(\mathbf{x}^\alpha f_i) = \mathrm{LM}(\mathbf{x}^\beta f_j) \text{ and } i < j \end{array}$$

*is called **Schreyer ordering** on F_1.*

Obviously the Schreyer ordering depends on the ordering $<$ on M and on the elements f_i. The following statement is an immediate consequence of the properties of $<$.

Lemma 1.2.8. *The Schreyer ordering is a module ordering.*

The concept of leading data can be extended to modules over polynomial rings.

Definition 1.2.9. *Let $M = R[\mathbf{x}]^m$ be an $R[\mathbf{x}]$-module with monomial ordering $<$ and consider an element*

$$f = \sum_{\alpha,i} a_\alpha^{(i)} \mathbf{x}^\alpha \mathbf{e}_i \in M \setminus \{0\}.$$

*Then we define the **support** of f, $\mathrm{supp}(f) = \{\mathbf{x}^\alpha \mathbf{e}_i \mid a_\alpha^{(i)} \neq 0\}$, and with $\mathbf{x}^\beta \mathbf{e}_j := \max \mathrm{supp}(f)$, the following symbols:*

$\mathrm{LM}(f) := \mathbf{x}^\beta \mathbf{e}_j$	*leading monomial of f,*
$\mathrm{LComp}(f) := \mathbf{e}_j$	*leading component of f,*
$\mathrm{LE}(f) := \beta$	*leading exponent of f,*
$\mathrm{LC}(f) := a_\beta^{(j)}$	*leading coefficient of f,*
$\mathrm{LT}(f) := a_\beta^{(j)} \mathbf{x}^\beta \mathbf{e}_j$	*leading term of f,*
$\mathrm{tail}(f) := f - \mathrm{LT}(f)$	*tail of f.*

We may consider the free module over the localization $R[\mathbf{x}]_<$, i.e., $M = \oplus_{i=1}^m R[\mathbf{x}]_< \cdot \mathbf{e}_i$. Since for any $f \in R[\mathbf{x}]_<$ there is a unit u such that $u \cdot f \in R[\mathbf{x}]$ and $\mathrm{LC}(u) = 1$, it is easy to see that for each $g \in M$ there is a unit v with $\mathrm{LC}(v) = 1$ such that $v \cdot g \in \oplus_{i=1}^m R[\mathbf{x}] \cdot \mathbf{e}_i$.

We may then use the leading data of $v \cdot g$, according to Def. 1.2.9, to define the leading data of g, which will again be independent from the choice of v.

Furthermore, Definition 1.1.27 extends naturally to modules:

Definition 1.2.10. *Let $M = (R[\mathbf{x}]_<)^m$ be a module and $<$ a module monomial ordering on $(R[\mathbf{x}])^m$, and $N \subset M$ any subset. We define the submodules*

$$\begin{aligned} \mathrm{L}(N) &= \langle \mathrm{LT}(f) \mid f \in N \rangle_{(R[\mathbf{x}])^m} && \text{leading module of } N, \\ \mathrm{LM}(N) &= \langle \mathrm{LM}(f) \mid f \in N \rangle_{(R[\mathbf{x}])^m} && \text{leading monomials module of } N, \end{aligned}$$

which are $R[\mathbf{x}]$-submodules of $(R[\mathbf{x}])^m$

1.3. Normal forms and standard bases

In the following let R be a noetherian ring, $<$ be a monomial ordering on $R[\mathbf{x}]$ and we set

$$A := R[\mathbf{x}]_<, M := \bigoplus_{i=1}^{m} A\mathbf{e}_i,$$

the free module of rank m over the localization of $R[\mathbf{x}]$ by $<$. Further we assume that a compatible module monomial ordering on $\mathrm{Mon}(M)$, also denoted by $<$, is given.

Definition 1.3.1. *Let $\mathcal{G}$ denote the set of all finite lists G with elements from M. We call a map*

$$\mathrm{NF} : M \times \mathcal{G} \to M, (f, G) \mapsto \mathrm{NF}(f \mid G)$$

*i. a **weak normal form** on M if, for all $G \in \mathcal{G}$,*

0) $\mathrm{NF}(0 \mid G) = 0$,

and, for all $f \in M$ and $G \in \mathcal{G}$,

1) $\mathrm{NF}(f \mid G) \neq 0 \implies \mathrm{LT}(\mathrm{NF}(f \mid G)) \notin \mathrm{L}(G)$.

2) $f \neq 0 \implies$ There exits a unit $u \in \mathrm{E}(R[\mathbf{x}]_<)$ with either $u \cdot f = \mathrm{NF}(f \mid G)$, or the remainder

$$r := u \cdot f - \mathrm{NF}(f \mid G) = \sum_{g \in G} a_g \cdot g, \quad a_g \in R[\mathbf{x}]_<, \tag{1.5}$$

has a standard representation (with respect to G), i. e. for all $g \in G$ with $a_g \cdot g \neq 0$ the relation $\mathrm{LM}(r) \geq \mathrm{LM}(a_g) \cdot \mathrm{LM}(g)$ holds.

*ii. a **normal form**, if we can choose $u = 1$ for every f and G in (1.5).*

*iii. **polynomial** if it is a weak normal form and whenever the input is polynomial, i. e. $f \in M \cap \bigoplus_{i=1}^m R[\mathbf{x}]\mathbf{e}_i$ and $G \subset \bigoplus_{i=1}^m R[\mathbf{x}]\mathbf{e}_i$, there exists a unit $u \in \mathrm{E}(R[\mathbf{x}]_<) \cap R[\mathbf{x}]$, such that $uf - \mathrm{NF}(f \mid G)$ has a standard representation with polynomials a_i, that is, $a_i \in R[\mathbf{x}]$.*

*iv. **reduced** if $t \notin \mathrm{L}(G)$ for all terms $t \neq 0$ of $\mathrm{NF}(f \mid G)$.*

Remark 1.3.2. *For global orderings we have* $\mathrm{E}(R[\mathbf{x}]_<) = \mathrm{E}(R)$. *Therefore in this case, the notions of weak normal form and normal form coincide.*

Remark 1.3.3. *Polynomial weak normal forms exists for arbitrary noetherian rings and are computable, given linear equations in* R *are solvable (Theorem 1.3.15).*

Proposition 1.3.4. *Let* NF *be a (weak) normal form,* $G \subset M$ *finite, and* $h = \mathrm{NF}(f \mid G)$ *the normal form of* $f \in M \setminus \{0\}$. *Then, with* r *as in (1.5),*

i. $h \neq 0 \implies \mathrm{LM}(f) \geq \mathrm{LM}(h)$,

ii. $r \neq 0 \implies \mathrm{LM}(f) \geq \mathrm{LM}(a_g) \cdot \mathrm{LM}(g)$, *for all* $g \in G$ *with* $a_g \cdot g \neq 0$,

iii. $r \neq 0 \implies \mathrm{LM}(f) \geq \mathrm{LM}(r)$,

iv. $r \neq 0 \implies \mathrm{LM}(r) = \max\{\mathrm{LM}(a_g) \cdot \mathrm{LM}(g) \mid g \in G\}$.

Proof. For showing ii. assume for a contradiction that $\mathrm{LM}(a_g) \cdot \mathrm{LM}(g) > \mathrm{LM}(f)$, for some $g \in G$ with $a_g \cdot g \neq 0$. The standard representation of r implies $\mathrm{LM}(r) \geq \mathrm{LM}(a_g) \cdot \mathrm{LM}(g)$, therefore $\mathrm{LM}(r) > \mathrm{LM}(f)$. Since $r = uf - h$ and $\mathrm{LM}(f) = \mathrm{LM}(uf)$ we obtain $\mathrm{LT}(h) = \mathrm{LT}(r) = \sum_{g \in G'} \mathrm{LT}(a_g) \cdot \mathrm{LT}(g)$ for some subset $G' \subset G$. Hence $\mathrm{LT}(h) \in \mathrm{L}(G)$, violating property 1) of weak normal forms.
iii. is an immediate consequence of ii. since r is the sum of all $a_g \cdot g, g \in G$.
If $r = 0$ then i. holds (with equality). On the other hand, if $r \neq 0$, then $\mathrm{LM}(h) = \mathrm{LM}(uf - r) \leq \mathrm{LM}(uf) = \mathrm{LM}(f)$ where we use iii.
Finally, to show iv. we note that

$$\begin{aligned}\max(\mathrm{LM}(a_g) \cdot \mathrm{LM}(g) \mid g \in G) &\geq \mathrm{LM}\left(\sum_{g \in G} a_g \cdot g\right) \\ &\geq \mathrm{LM}(a_g) \cdot \mathrm{LM}(g) \text{ for all } g \in G,\end{aligned}$$

where the last inequality is again due to the standard representation of r. □

Having computed a (weak) normal form $\mathrm{NF}(f \mid G) = h_1\mathbf{e}_1 + \cdots + h_m\mathbf{e}_m$, we may sometimes even require $\mathrm{LT}(h_i\mathbf{e}_i) \notin \mathrm{L}(G)$ for all i with $h_i \neq 0$. Such a (weak) normal form is called **component-reduced** and is computed by Algorithm 1.

Lemma 1.3.5. *Let* $M = \bigoplus_{i=1}^m A \cdot \mathbf{e}_i$ *be a free* A*-module with monomial ordering* $<$. *Then Algorithm 1 terminates after at most* $m + 1$ *steps and computes a component-reduced (weak) normal form on* M.

Proof. We set $E := \{\mathbf{e}_1, \ldots, \mathbf{e}_m\}$ and define

$$\begin{aligned}\mathbf{x}_1 \cdot \mathbf{e} <_C \mathbf{x}_2 \cdot \mathbf{f} \quad :\Longleftrightarrow \quad & \mathbf{e} \in C \wedge \mathbf{f} \notin C \\ & \text{or } (\mathbf{e}, \mathbf{f} \in C \vee \mathbf{e}, \mathbf{f} \notin C) \wedge \mathbf{x}_1 \cdot \mathbf{e} < \mathbf{x}_2 \cdot \mathbf{f} \\ \text{and the set} \quad G_C := \quad & \{g \in G \mid \mathrm{LComp}(g) \notin C\},\end{aligned}$$

Algorithm 1 Computing a component-reduced (weak) normal form

Input: $M = \bigoplus_{i=1}^{m} A \cdot \mathbf{e}_i$ a free A-module with module monomial ordering $<$
Input: for each $C \subset \{\mathbf{e}_1, \ldots, \mathbf{e}_m\}$ a normal form $\mathrm{NF}(. \mid .)_C$ for the module monomial ordering $<_C$ as defined in the proof of Lemma 1.3.5
Input: $G \subset M$ a finite subset of module elements
Input: $f \in M$ an arbitrary module element
Output: h a component-reduced (weak) normal form of f

```
C := ∅
h := f
while h ∉ ⊕_{e∈C} A·e do
  h := NF_C(h | {g ∈ G | LComp(g) ∉ C})
  e := LComp_C(h)
  C := C ∪ {e}
end while
return h
```

for all $C \subset E$, $\mathbf{e}, \mathbf{f} \in E$ and $\mathbf{x}_1, \mathbf{x}_2$.

First note that the monomial orderings $<_\emptyset$, $<$ and $<_E$ coincide, and that

$$\mathbf{x}_1 \cdot \mathbf{e} < \mathbf{x}_2 \cdot \mathbf{e} \iff \mathbf{x}_1 \cdot \mathbf{e} <_C \mathbf{x}_2 \cdot \mathbf{e} \text{ and} \tag{1.6}$$

$$\mathbf{e} \in C, \mathbf{f} \notin C \implies \mathbf{x} \cdot \mathbf{e} <_C \mathbf{x} \cdot \mathbf{f}, \tag{1.7}$$

$$g \in G_c \implies \mathrm{LT}(g) = \mathrm{LT}_C(g) \tag{1.8}$$

for all $C \subset E$, $\mathbf{e}, \mathbf{f} \in E$ and $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2$. If we consider the leading monomial or term of an element $a \cdot \mathbf{e}$ with $a \in A$ we will write $\mathrm{LM}(a \cdot \mathbf{e})$ instead of $\mathrm{LM}_C(a \cdot \mathbf{e})$ to emphasize the independence (1.6) of the leading data from the chosen ordering implied by C.

To prove termination consider a loop in which the computed component $\mathbf{e}$ is already an element of C. Then $\mathbf{e}$ is the leading component (with respect to $<_C$) of the newly computed $h' = \mathrm{NF}_C(h \mid G_C)$, i.e. $\mathrm{LM}_C(h') = \mathbf{x}_1 \cdot \mathbf{e}$ for some monomial $\mathbf{x}_1$. If h' had a term $c \cdot \mathbf{x}_2 \cdot \mathbf{f}$ with $\mathbf{f} \notin C$ then $\mathbf{x}_2 \cdot \mathbf{f} >_C \mathbf{x}_1 \cdot \mathbf{e}$ by (1.7), a contradiction. Consequently, $h' \in \bigoplus_{\mathbf{e} \in C} A \cdot \mathbf{e}$, and the while loop terminates after at most $m + 1$ steps.

In order to prove correctness note first that there is nothing to show for the case that one of the normal form computations yields zero, implying that Algorithm 1 returns zero.

Thus we consider the following data representing one turn in the while loop:

$$h' = \mathrm{NF}_C(h \mid G_C = \{g \in G \mid \mathrm{LComp}(g) \notin C\}) \tag{1.9}$$

$$\mathbf{e}' = \mathrm{LComp}_C(h') \tag{1.10}$$

$$h = \sum_{\mathbf{e} \in E} h_{\mathbf{e}} \cdot \mathbf{e} \quad ; \quad h' = \sum_{\mathbf{e} \in E} h'_{\mathbf{e}} \cdot \mathbf{e} \tag{1.11}$$

and assume $h, h' \neq 0$. Let moreover, for each component $\mathbf{e}$ which belongs to the final set C, $C_{\mathbf{e}}$ denote the set C at the beginning of the loop in which $\mathbf{e}$ is inserted into C.

We shall show the following claims

$$\begin{aligned} &i) \quad \mathbf{e}' \notin C \implies \mathrm{LT}\left(h'_{\mathbf{e}'} \cdot \mathbf{e}'\right) \left[\overset{(1.6)}{=} \mathrm{LT}_C\left(h'_{\mathbf{e}'} \cdot \mathbf{e}'\right) \overset{(1.10)}{=} \mathrm{LT}_C\left(h'\right) \right] \notin \mathrm{L}\left(G\right) \text{ and} \\ &ii) \quad \mathbf{e} \in C \implies \mathrm{LT}_{C_{\mathbf{e}}}\left(h'\right) = \mathrm{LT}\left(h'_{\mathbf{e}} \cdot \mathbf{e}\right) = v_{\mathbf{e}} \cdot \mathrm{LT}_{C_{\mathbf{e}}}\left(h\right) \text{ with some } v_{\mathbf{e}} \in \mathrm{E}\left(A\right), \end{aligned}$$

where $\mathbf{e}'$ in i) refers to one specific loop, see (1.10), and ii) allows for any component in any set C during the entire while loop.

For the final value $\hat{h}$ of h we have $\hat{h} \in \bigoplus_{\mathbf{e} \in C} A \cdot \mathbf{e}$, i. e. if $\hat{h}_{\mathbf{e}} \neq 0$ for some $\mathbf{e} \in E$, then $\mathbf{e} \in C$. Claim ii) then states that the leading terms (with respect to $<_{C_{\mathbf{e}}}$) of $\hat{h}$ and its predecessors back to the loop in which $\mathbf{e}$ was inserted into C are the same up to units. But for the loop in which $\mathbf{e}$ was inserted, claim i) ensures that this leading term is not in $\mathrm{L}\left(G\right)$. Hence, the claims suffice to prove that Algorithm 1 returns indeed a component-reduced (weak) normal form.

To prove claim i) note that the (weak) normal form (see Definition 1.3.1 i.) ensures that $\mathrm{LT}\left(h'_{\mathbf{e}'} \cdot \mathbf{e}'\right) = \mathrm{LT}_C\left(h'\right) \notin \mathrm{L}_C\left(G_C\right)$. For a contradiction, assume $\mathrm{LT}_C\left(h'\right) \in \mathrm{L}\left(G\right)$.

$$\begin{aligned} &\implies \mathrm{LT}_C\left(h'\right) \in \mathrm{L}\left(G\right) \cap A \cdot \mathbf{e}' = \mathrm{L}\left(G_C\right) \cap A \cdot \mathbf{e}', \text{ since } \mathbf{e}' \notin C, \\ &\implies \mathrm{LT}_C\left(h'\right) \in \mathrm{L}_C\left(G_C\right) \cap A \cdot \mathbf{e}' \subset \mathrm{L}_C\left(G_C\right), \end{aligned}$$

where the last implication follows from the fact that for $g \in G_C, \mathrm{LT}\left(g\right) = \mathrm{LT}_C\left(g\right)$, see (1.7).

We will prove claim ii) by induction on the size of C. For $C = \emptyset$ there is nothing to show. Due to property ii) of (weak) normal forms we have a standard representation for $u \cdot h - h' = \sum_{g \in G_C} a_g \cdot g$ with some unit $u \in \mathrm{E}\left(A\right)$. Note that $a_g \in A$ and therefore the leading monomial of a_g is defined by the ring monomial ordering and not by the module monomial ordering. Now consider a component $\mathbf{f} \in C$. With the arguments (1.12) and (1.13) proved below, we have

$$\begin{aligned} \mathrm{LM}\left(h_{\mathbf{f}} \cdot \mathbf{f}\right) &= \mathrm{LM}_{C_{\mathbf{f}}}\left(h\right) && \text{by induction hypothesis,} \\ &>_{C_{\mathbf{f}}} \mathrm{LM}_C\left(h\right) && (1.12) \\ &= \mathrm{LM}_C\left(u \cdot h\right) \\ &\geq_{C_{\mathbf{f}}} \mathrm{LM}_C\left(\sum a_g \cdot g\right) && (1.13) \\ &= \max_{g \in G_C} (\mathrm{LM}\left(a_g\right) \cdot \mathrm{LM}_C\left(g\right)) && \text{by Proposition 1.3.4 iv.,} \\ &= \max_{g \in G_C} (\mathrm{LM}\left(a_g\right) \cdot \mathrm{LM}_{C_{\mathbf{f}}}\left(g\right)) && \text{since } \mathrm{LComp}\left(g\right) \notin C \supset C_{\mathbf{f}}, \\ &\geq_{C_{\mathbf{f}}} \mathrm{LM}_{C_{\mathbf{f}}}\left(\sum a_g \cdot g\right) \end{aligned}$$

Because of $h' = u \cdot h - \sum a_g \cdot g$ and the just deduced relation $\mathrm{LM}_{C_{\mathbf{f}}}\left(h\right) >_{C_{\mathbf{f}}} \mathrm{LM}_{C_{\mathbf{f}}}\left(\sum a_g \cdot g\right)$ we know that $\mathrm{LT}_{C_{\mathbf{f}}}\left(h'\right) = v_{\mathbf{f}} \cdot \mathrm{LT}_{C_{\mathbf{f}}}\left(h\right)$. In particular, $\mathrm{LComp}_{C_{\mathbf{f}}}\left(h'\right) = \mathrm{LComp}_{C_{\mathbf{f}}}\left(h\right) = \mathbf{f}$. Hence $\mathrm{LT}_{C_{\mathbf{f}}}\left(h'\right) = \mathrm{LT}\left(h'_{\mathbf{f}} \cdot \mathbf{f}\right)$. This shows ii).

It remains to show (1.12) and (1.13). To justify (1.12) we need to show that the leading corresponding leading monomials differ, i. e. $\mathrm{LM}_{C_{\mathbf{f}}}\left(h\right) \neq \mathrm{LM}_C\left(h\right)$. Therefore

assume $\mathrm{LM}_{C_\mathbf{f}}(h) = \mathrm{LM}_C(h)$. Since $\mathrm{LComp}_{C_\mathbf{f}}(h) = \mathbf{f}$ we obtain $\mathrm{LComp}_C(h) = \mathbf{f} \in C$ and hence $h \in \oplus_{\mathbf{e}\in C} A \cdot \mathbf{e}$, a contradiction to the entry condition of the while loop.

Now consider (1.13) and $\mathrm{LM}_C(u \cdot h)$. Proposition 1.3.4 yields

$$\begin{aligned}\mathrm{LM}_C(u \cdot h) &\geq_C \mathrm{LM}_C\left(\sum a_g \cdot g\right)\\ &= \max_{g\in G_C}\{\mathrm{LM}(a_g) \cdot \mathrm{LM}_C(g)\}\end{aligned}$$

Due to the while condition the leading component $\mathrm{LComp}_C(u \cdot h)$ is not in C. Furthermore, since the latter equation holds, the leading component of the sum equals the leading component of some $g \in G_C$, i. e. is also not in C. Hence we can replace $<_C$ by $<_D$ for any subset $D \subset C$, in particular by $C_\mathbf{f}$.

□

Definition 1.3.6. *Let $I \subset M$ be a submodule. A finite set $G \subset R$ is called **standard basis** (SB) of I if*

$$G \subset I, \text{ and } \mathrm{L}(I) = \mathrm{L}(G).$$

*That is, G is a standard basis if the leading terms of G generate the leading module of I. Further G is called a **strong standard basis** (sSB), if for any $f \in I\backslash\{0\}$ there exists a $g \in G$ satisfying $\mathrm{LT}(g) \,|\, \mathrm{LT}(f)$. Of course, any sSB is an SB. If $<$ is global we call standard bases also **Gröbner bases**.*

Remark 1.3.7.

1. *If R is a field, then $\mathrm{L}(I) = \mathrm{LM}(I)$. But due to non invertible coefficients in the general case only $\mathrm{L}(I) \subset \mathrm{LM}(I)$ holds.*
2. *If R is a field, then $\mathrm{LT}(f) \in \mathrm{L}(G)$ if and only if $\mathrm{LT}(g) \,|\, \mathrm{LT}(f)$ for some $g \in G$ and hence any standard basis is strong. For arbitrary rings R this is not true.*

Example 1.3.8.

1. *Consider $I = \langle 2x\rangle \subset \mathbb{Z}[x]$. Then $\mathrm{L}(I) = \langle 2x\rangle$ and $\mathrm{LM}(I) = \langle x\rangle$. Since $x \notin \langle 2x\rangle$ it is impossible to achieve $\langle G\rangle = \mathrm{LM}(I)$ with $G \subset I$. But for $G = \langle 2x\rangle$ we have $\mathrm{L}(I) = \mathrm{L}(G)$ and hence G is a (strong) Gröbner basis of I.*
2. *Consider again $R = \mathbb{Z}[x]$ and $I = \langle x\rangle$. Then $G = \langle 2x, 3x\rangle$ is clearly a SB of I but not a strong one.*

Lemma 1.3.9. *Let $I \subset M$ be a submodule, $G \subset I$ a standard basis of I and $\mathrm{NF}(-\mid G)$ a weak normal form on M with respect to G.*

1. *For any $f \in M$ we have $f \in I$ if and only if $\mathrm{NF}(f \mid G) = 0$.*
2. *If $J \subset M$ is a submodule with $I \subset J$, then $\mathrm{L}(I) = \mathrm{L}(J)$ implies $I = J$.*
3. *$I = \langle G\rangle_{R[\mathbf{x}]_<}$, that is, the standard basis generates I as an $R[\mathbf{x}]_<$-module.*
4. *If $\mathrm{LM}(G) = \mathrm{L}(G)$ and $\mathrm{NF}(-\mid G)$ is a reduced normal form, then it is unique.*

Proof. The ideals $\mathrm{L}(G)$ and $\mathrm{L}(I)$ coincide since G is a standard basis of I. Let $uf - \mathrm{NF}(f \mid G) = r \in \langle G \rangle \subset I$. At first assume that $f \in I$. Then $\mathrm{NF}(f \mid G) \in I$ and hence $\mathrm{L}(\mathrm{NF}(f \mid G)) \in \mathrm{L}(G)$. It has to be $\mathrm{NF}(f \mid G) = 0$ as otherwise property (1) of weak normal forms would be violated. Now assume $\mathrm{NF}(f \mid G) = 0$, then $f = u^{-1} \cdot r \in I$ as stated.

Now let $f \in J \backslash I$ and therefore $\mathrm{NF}(f \mid G) = uf - r \neq 0$ by the first statement. Property (2) of weak normal forms implies the following equality $\mathrm{L}(\mathrm{NF}(f \mid G)) \notin \mathrm{L}(G) = \mathrm{L}(I) = \mathrm{L}(J)$. But $\mathrm{NF}(f \mid G) \in J$ since $uf \in J$ and $r \in I \subset J$, a contradiction.

For the third statement observe that $\langle G \rangle \subset I$ and $\mathrm{L}(I) \supset \mathrm{L}(\langle G \rangle) \supset \mathrm{L}(G) = \mathrm{L}(I)$ and apply the second.

Now assume $\mathrm{L}(G) = \mathrm{LM}(G)$ and let h, h' be reduced normal forms if the same polynomial f, i. e. no term of h or h' is divisible by any term of $\mathrm{L}(G) = \mathrm{LM}(G)$. If $h - h' \neq 0$ then $\mathrm{LT}(h - h') \in \mathrm{LM}(G)$. This is a contradiction, since $\mathrm{LM}(h - h')$ is a monomial of h or h'. □

Example 1.3.10. *We may not drop the requirement* $\mathrm{LM}(G) = \mathrm{L}(G)$ *in the last statement. Take for instance* $G = \{2x^2\} \subset \mathbb{Z}[x]$ *and* $I = \langle G \rangle$. *Further take* $f = 3x^2, h = 3x^2, h' = x^2$. *Then* h *and* h' *are normal forms of* f *and further* $h - h' = 2x^2 \in \mathrm{L}(G)$, *but neither* $3x^2$ *nor* x^2 *are in* $\mathrm{L}(G)$.

However, if we have a useful definition of reduction for coefficients we can achieve uniqueness (in case of Dedekind domains see [1]). The is possible for a wide range of rings, i.e. the integers, $\mathbb{Z}/m$, *and more.*

Remark 1.3.11. *Let* R *be a principal ideal domain and* h *be a non-zero normal form of an element* $f \in M$. *Further let* $G_{\mathrm{LM}(h)} = \{g \in G \mid \mathrm{LM}(g) \mid \mathrm{LM}(h)\}$. *A canonical choice for the leading coefficient of a normal form up to multiplication with a unit is a generator* a *of the ideal in* R *generated by* $\mathrm{LC}(h)$ *and* $\{\mathrm{LC}(g) \mid g \in G_{\mathrm{LM}(h)}\}$, *i.e.* $\langle a \rangle_R = \left\langle \mathrm{LC}(h), \mathrm{LC}(g) \mid g \in G_{\mathrm{LM}(h)} \right\rangle$.

Now we introduce an algorithm to compute a polynomial weak normal form, given the ability to solve an arbitrary linear equation in the coefficient ring R. To ensure termination we introduce the concept of the ecart of a polynomial.

Definition 1.3.12. *Let* $f \in R[\mathbf{x}]^s_< \backslash \{0\}$. *The* ecart *is defined by*

$$\mathrm{ecart}\, f = \deg f - \deg \mathrm{LM}(f).$$

We define a monomial ordering $<_h$ on $R[t, \mathbf{x}]^s_<$ by

$$t^p \mathbf{x}^\alpha <_h t^q \mathbf{x}^\beta :\Longleftrightarrow \; p + |\alpha| < q + |\beta| \text{ or } \left(p + |\alpha| = q + |\beta| \text{ and } \mathbf{x}^\alpha < \mathbf{x}^\beta\right),$$

i. e. the monomials are ordered first by there total degree and then by their $\mathbf{x}$ part according to $<$. The ordering $<_h$ is therefore a well-ordering regardless of $<$ as there are only finitely many monomials of a given total degree.

Algorithm 2 Computing a normal form over coefficient rings (NF-MGPS)

Input: R a ring where linear equations are solvable
Input: $M = \bigoplus_{i=1}^{m} R[\mathbf{x}]\mathbf{e}_i$ a free R-module
Input: $>$ a (module) monomial ordering on M
Input: $G \subset M$ a finite subset of module elements
Input: $f \in M$ an arbitrary module element
Output: return value is a weak normal form of f with respect to G and $<$.

```
T := {g^h | g ∈ G}
while f ≠ 0 and t^α LT(f) ∈ L(T) for some α do
  choose α minimal such that t^α LT(f) ∈ L(T) holds
  t^α LT(f) = Σ_{h∈T} c_h x^{β_h} t^{γ_h} LT(h)
                with x^{β_h} t^{γ_i} LM(h) = t^α LM(f) for all h with c_h LC(h) ≠ 0
  if α > 0 then
    T := T ∪ {f^h}
  end if
  f := f^h − Σ_{h∈T} c_i x^{β_i} t^γ h
  f := f(x, 1)
end while
return f
```

Remark 1.3.13. *Let $f^h \in R[t,\mathbf{x}]^s_{<_h}$ be the homogenization of f with respect to a new variable t, that is, each term in f will be complemented by a appropriate power of t such that all terms have degree* $\deg(f)$. *Then* $\mathrm{LT}\left(f^h\right) = t^{\mathrm{ecart}\, f}\,\mathrm{LT}\left(f\right)$.

Definition 1.3.14. *Let R be a ring. We say that* ***linear equations are solvable*** *in R if, for given elements $a_1,\ldots,a_m,b \in R$, $m \in \mathbb{N}$, the following data is computable.*

1. *A finite generating set $S \subset R^m$ of the syzygies*

$$\langle S\rangle_R = \mathrm{Syz}\left(a_1,\ldots,a_m\right) = \{x \in R^m \mid x_1\cdot a_1 + \cdots + x_m \cdot a_m = 0\}$$

2. *A decision whether there exists a special solution $(x_1,\ldots,x_m) \in R^m$ with*

$$x_1\cdot a_1 + \cdots + x_m\cdot a_m = b$$

and, if it exists, the element $(x_1,\ldots,x_m)$.

Theorem 1.3.15. *Let R be a ring where special solutions of linear equations are computable i.e. R satisfies 2). Then Algorithm 2 terminates and computes a polynomial (weak) norm form.*

Proof. Denote by T_ν the set T in the ν-th run. Since the ring $R[\mathbf{x},t]$ is noetherian, the increasing chain of submodules $\mathrm{L}\left(T_\nu\right) \subset R[\mathbf{x},t]^s$ will stabilize at some point. Then also

T_ν must stabilize, as the minimal α to solve

$$t^\alpha \operatorname{LT}(f) = \sum_{i=1}^{s} c_i \, \mathbf{x}^{\alpha_i} \operatorname{LT}(g_i)$$

will be zero and hence no polynomials will be added to T. After this point every reduction of f will be done with $\alpha = 0$. Further $<_h$ is a well ordering on the monomials $\operatorname{Mon}(\mathbf{x}, t) \times \{\mathbf{e}_1, \ldots, \mathbf{e}_s\}$, therefore we can remove the leading term of f only finitely many times, before either reaching $f = 0$ or $f \notin \operatorname{L}(T)$.

Now let f_ν denote the polynomial f in the ν-th run and f_N the final one. We have the following recursive definition of f_ν

$$f_\nu = f_{\nu-1} - \sum_i m_i^\nu g_i^\nu, \text{ such that } \operatorname{LM}(f_\nu) < \operatorname{LM}(f_{\nu-1})$$

and m_i^ν denotes the corresponding $c_i \mathbf{x}^{\beta_i}$. We prove by induction on ν that every $u_\nu f - f_\nu$ has a representation $\sum_i a_i g_i$ with $u_\nu \in \operatorname{E}(R[\mathbf{x}]_<) \cap R[\mathbf{x}]$, $g_i \in G$ and $\operatorname{LM}(a_i) \cdot \operatorname{LM}(g_i) \leq \operatorname{LM}(f - f_\nu)$. Therefore f has a standard representation with respect to G, i. e. the one of $f - f_N$. First see, that $f - f_1 = \sum_i m_i^1 g_i^1$ with $\operatorname{LM}(f) = \operatorname{LM}(m_i^1) \cdot \operatorname{LM}(g_i^1) > \operatorname{LM}(f_1)$ and $g_i \in G$ is a representation of the required type. Now assume for $k \leq \nu$ and $u_k \in S_<$ the polynomial $u_k f - f_k$ has a standard representation $\sum a_i g_i$ with $g_i \in G$ (we omit indices, as they do not contribute). We can write

$$\begin{aligned}
f_{\nu+1} &= f_\nu - \sum_i m_i^\nu g_i^\nu \\
&= u_\nu f - \sum a_i g_i - \sum_i m_i^\nu g_i^\nu \qquad \text{representation of } u_\nu f - f_\nu \\
&= u_\nu f - \sum a_i g_i - \sum_{i, g_i^\nu \in G} m_i^\nu g_i^\nu - \sum_{i, g_i^\nu \notin G} m_i^\nu g_i^\nu
\end{aligned}$$

Since each $g_i^\nu \notin G$ were added during the algorithm there are ν_i, such that

$$g_i^\nu = f_{\nu_i} = u_{\nu_i} f - \sum a_j g_j, \qquad \nu_i < \nu.$$

We deduce

$$\begin{aligned}
f_{\nu+1} &= u_\nu f - \sum_i m_i^\nu f_{\nu_i} - \sum a_i g_i \\
&= u_\nu f - \sum_i \left(m_i^\nu u_{\nu_i} f - \sum m_i^\nu a_j g_j \right) - \sum a_i g_i \\
&= u_\nu f - f \sum_i m_i^\nu u_{\nu_i} - \sum a_i g_i \\
&= \left(u_\nu - \sum_i m_i^\nu u_{\nu_i} \right) f - \sum a_i g_i.
\end{aligned}$$

Therefore it remains to show, that $u_{\nu+1} = u_\nu - \sum m_i^\nu u_{\nu_i}$ is a unit, i.e. its leading monomial is 1 and its leading coefficient is a unit in C. Since $\mathrm{LM}(m_i^\nu)\,\mathrm{LM}(f_{\nu_i}) = \mathrm{LM}(m_i^\nu f_{\nu_i}) = \mathrm{LM}(m_i^\nu g_i^\nu) = \mathrm{LM}(f_\nu) < \mathrm{LM}(f_{\nu_i})$ we deduce $\mathrm{LM}(m_i^\nu) < 1$ and therefore also $\mathrm{LM}(m_i^\nu u_{\nu_i}) < 1 = \mathrm{LM}(u_{\nu_i})$ as $\mathrm{LC}(u_{\nu_i})$ is a unit. In particular $\mathrm{LC}(u_{\nu+1}) = \mathrm{LC}(u_\nu - \sum m_i^\nu u_{\nu_i}) = \mathrm{LC}(u_\nu) = \mathrm{LC}(u_1) = 1$, i.e. a unit. □

Remark 1.3.16. *Solving linear equations during the normal form computation is not necessary if G is a strong standard basis (see Algorithm 3).*

Algorithm 3 Calculating a normal form using a strong standard basis

Input: R a ring where divisibility is decidable
Input: $M = \bigoplus_{i=1}^m R[\mathbf{x}]\mathbf{e}_i$ a free R-module
Input: $>$ a (module) monomial ordering on M
Input: $G \subset M$ a finite subset of module elements
Input: $f \in M$ an arbitrary module element
Output: return value is a weak normal form of f

```
T := G
while f ≠ 0 and ∅ ≠ T' = {g ∈ T : LT(g) | LT(f)} do
  Select g ∈ T' with ecart g = min
  Let LT(f) = c x^α · LT(g) with c · LC(g) ≠ 0
  if ecart g > 0 then
    T := T ∪ {f}
  end if
  f := f − c x^α · g_i
end while
return f
```

1.3.1. Standard bases

As in the previous section let R be a ring and M the free $R[\mathbf{x}]_<$–module $\oplus_{i=1}^m R[\mathbf{x}]_< \mathbf{e}_i$.

Theorem 1.3.17. *Let $I \subset M$ be a submodule and $G = \{g_1, \ldots, g_s\} \subset I \cap \oplus_{i=1}^m R[\mathbf{x}]\mathbf{e}_i$. Further let $\mathrm{NF}(- \mid G)$ be a weak normal form on the module M with respect to G. Then the following statements are equivalent:*

1. *G is a standard basis of I.*
2. *$\mathrm{NF}(f \mid G) = 0$ for all $f \in I$.*
3. *Each $f \in I$ has a standard representation with respect to G.*

Proof. The implication 1. ⇒ 2. was already proven in Lemma 1.3.9. Property (2) of weak normal forms explicitly states the existence of a standard representation for all f with $\mathrm{NF}(f \mid G) = 0$.

Now assume that every element $f \in I$ has a standard representation with respect to G. We have to show that $\mathrm{L}(G) = \mathrm{L}(I)$ since $G \subset I$ by definition. Therefore let $f \in I$ and $f = \sum a_i g_i$ be a standard representation of f, i. e. $\mathrm{LM}(a_i)\,\mathrm{LM}(g_i) \leq \mathrm{LM}(f)$. Hence the leading term $\mathrm{LT}(f) = \sum_{j \in J} \mathrm{LT}(a_j)\,\mathrm{LT}(g_j)$ with $J = \{j \mid \mathrm{LM}(a_j)\,\mathrm{LM}(g_j) = \max_i \mathrm{LM}(a_i)\,\mathrm{LM}(g_i)\}$. Therefore $\mathrm{LT}(f) \in \langle \mathrm{LT}(g_i) \rangle = \mathrm{L}(G)$ and, since f was arbitrary, $\mathrm{L}(G) = \mathrm{L}(I)$ and G is a standard basis. □

Algorithm 4 Computes a standard basis of I
(given a finite generating set of the module of syzygies)

Input: R a ring
Input: M a free R-module
Input: $>$ a (module) monomial ordering on M
Input: NF a component-reduced weak normal form on M
Input: $\mathrm{Gen}_{\mathrm{fin}}(\mathrm{Syz}(\cdot))$ over R is computable
Input: $I = \{f_1, \ldots, f_l\} \subset M$ a finite set of module elements
Output: $S = \{s_1, \ldots, s_r\}$ is a standard basis of the submodule generated by I

```
S := I as a ordered set, for the purpose of syzygies
P := Gen_fin(Syz(LT(s_i) | s_i ∈ S)) a finite generating set
while P ≠ ∅ do
  choose (g_1, ..., g_r) ∈ P, the cardinality of S is r
  P := P\{(g_1, ..., g_r)}
  h := NF(g_1 s_1 + ··· + g_r s_r | S)
  if h ≠ 0 then
    S := S ∪ {h}
    P := (P × {0}) ∪ {h ∈ Gen_fin(Syz(LT(s_i) | s_i ∈ S)) | h_r ≠ 0}
  end if
end while
return S
```

Lemma 1.3.18. *Let R be a ring where linear equation are solvable. Further assume, that for given terms $t_i = a_i \cdot \mathbf{x}_i$, a finite generating set $\mathrm{Gen}_{\mathrm{fin}}(t_i) \subset R$ for the syzygies is computable. Then Algorithm 4 terminates and computes the standard basis of I.*

Proof. Each time h is added to S the leading module $\mathrm{L}(S)$ is enlarged. Since R is noetherian this can only happen finitely many times. Hence, after S becomes stationary h will be zero for every remaining element of P and the algorithm terminates.

Obviously, $\langle I \rangle = \langle S \rangle$. By Theorem 1.3.17 it is sufficient to show

$$\mathrm{NF}\left(\sum_{i=1}^{k} a_i s_i \,\middle|\, S \right) = 0$$

for all $a_i \in R[\mathbf{x}]$. Assume $\mathrm{NF}\left(\sum_{i=1}^{k} a_i s_i \,\middle|\, S \right) = h \neq 0$. By Definition 1.3.1, $h = \sum_{i=1}^{k} b_i s_i$. Further $\sum_{i=1}^{k} \mathrm{LT}(b_i)\,\mathrm{LT}(s_i) = 0$ holds since $\mathrm{LT}(h) \notin \mathrm{L}(S)$, i. e. the leading

terms of the elements b_i defines a syzygy of the leading term of s_i. The syzygy is generate by elements which were at some time in the set P. But all leadings terms which can be generated by elements of P are in $\mathrm{L}(S)$ and hence $h = 0$. □

For a more rigorous proof see Theorem 1.4.4.

1.4. Standard basis computation and the syzygy theorem

In this chapter we prove the correctness and termination of Algorithm 5. Further we introduce notations which allow easy specializations to different coefficient rings.

Let R be a commutative noetherian ring with one where linear equations are solvable, $R[\mathbf{x}]$ the polynomial ring in n variables over R with monomial ordering $<$. The localization of $R[\mathbf{x}]$ with respect to $<$ will be called $A := R[\mathbf{x}]_<$. Further let $\bigoplus_{i=1}^m R[\mathbf{x}]\mathbf{e}_i$ be the free $R[\mathbf{x}]$-module of rank m and $<$ a module ordering compatible with the monomial ordering on $R[\mathbf{x}]$. We fix a set of polynomial generators $G = \{f_1, \ldots, f_l\} \subset \bigoplus_{j=1}^m R[\mathbf{x}] \cdot \mathbf{e}_j$ of a submodule $I \subset \bigoplus_{i=1}^m A \cdot \mathbf{e}_i =: M$.

We construct a set of syzygies S sufficient for standard basis computation. This set also leads to a standard basis of the syzygy submodule $\mathrm{Syz}(G) \subset F_1 = \bigoplus_{j=1}^l R[\mathbf{x}] \cdot \epsilon_j$ with respect to the Schreyer ordering. We quickly recall the ordering:

Definition 1.4.1. *Let $M = \bigoplus_{i=1}^m A \cdot \mathbf{e}_i$, $F_1 = \bigoplus_{i=1}^l A\,\epsilon_i$ and $f_1, \ldots, f_l \in M$. The ordering $<_1$ on F_1 given by*

$$\begin{aligned} \mathbf{x}^\alpha \mathbf{e}_i <_1 \mathbf{x}^\beta \mathbf{e}_j \Longleftrightarrow & \mathrm{LM}(\mathbf{x}^\alpha f_i) < \mathrm{LM}(\mathbf{x}^\beta f_j) \quad \textit{or} \\ & \mathrm{LM}(\mathbf{x}^\alpha f_i) = \mathrm{LM}(\mathbf{x}^\beta f_j) \quad \textit{and } i < j \end{aligned}$$

*is called the **Schreyer ordering** on F_1 and depends on the ordering $<$ on M and the elements f_i.*

We denote by φ the substitution homomorphism

$$\begin{aligned} \varphi : F_1 := \bigoplus_{i=1}^l A\,\epsilon_i &\longrightarrow M \\ \epsilon_i &\mapsto f_i \end{aligned}$$

The image of φ is the submodule I generated by $G = \{f_1, f_2, \ldots, f_l\}$. The kernel of φ is the syzygy module of $(f_1, f_2, \ldots, f_l)$.

We will construct a set which is a standard basis of the syzygies and proves that G is a standard basis of the submodule I given that certain normal forms are zero. At first consider for each $k = 1, 2, \ldots, l$ the set of all possible least common multiples involving only leading monomials of the first k elements from G, i. e.

$$\begin{aligned} C_k &:= \{\mathrm{lcm}(\mathrm{LM}(f_i) \mid i \in J) \mid k \in J \subset [k] \wedge \forall i \in J : \mathrm{LComp}(f_k) = \mathrm{LComp}(f_i)\} \\ [k] &:= \{1, 2, \ldots, k\}. \end{aligned}$$

We define the following derived sets for $m \in C_k$:

$$J_{k,m} := \{i \in [k] \mid \mathrm{LM}(f_i) \text{ divides } m\} \tag{1.14}$$

$$\mathrm{cSyz}_{k,m} := \{c \in \mathrm{Syz}_{\mathrm{Gen}}(\mathrm{LC}(f_i) \mid i \in J_{k,m}) \mid c_k \neq 0\} \subset \bigoplus_{j=1}^{l} R \cdot \epsilon_j \tag{1.15}$$

$$\mathrm{Syz}_k := \left\{ \sum_{i \in J_{k,m}} \frac{c_i \cdot m}{\mathrm{LM}(f_i)} \cdot \epsilon_i \;\middle|\; m \in C_k, c \in \mathrm{cSyz}_{k,m} \right\} \subset \bigoplus_{j=1}^{l} R[\mathbf{x}] \cdot \epsilon_j \tag{1.16}$$

For each $s \in \mathrm{Syz}_k$ we choose a standard representation of $u_s \cdot \varphi(s) - \mathrm{NF}(\varphi(s) \mid G) = \sum_{i=1}^{l} a_i f_i$ with $a_1, a_2, \ldots, a_l$ in $R[\mathbf{x}]_<$.

$$S_k := \left\{ u_s \cdot s - \sum_{i=1}^{l} a_i \epsilon_i \;\middle|\; s \in \mathrm{Syz}_k \right\} \tag{1.17}$$

Remark 1.4.2.

- *If* $c = \sum_{j=1}^{l} c_j \cdot \epsilon_j \in \mathrm{cSyz}_{k,m}$ *then* $\sum_{j=1}^{l} c_j \cdot \mathrm{LC}(f_j) = 0$, *i. e. c is a syzygy of the leading coefficients of* $f_1, f_2, \ldots, f_l$.
- *If* $s = \sum_{j=1}^{l} s_j \cdot \epsilon_j \in \mathrm{Syz}_k$ *then* $\sum_{j=1}^{l} s_j \cdot \mathrm{LT}(f_j) = 0$, *i. e. s is a syzygy of the leading terms of* $f_1, f_2, \ldots, f_l$.
- $\varphi(u_s \cdot s - \sum_{i=1}^{l} a_i)$ *is a (weak) normal form of* $\varphi(s)$ *with respect to* G.

Note that $\bigcup_{k=1}^{l} \mathrm{Syz}_k$ *corresponds to a "homogeneous generating set" for the syzygies of* $\mathrm{LT}(f_1), \mathrm{LT}(f_2), \ldots, \mathrm{LT}(f_l)$ *in the sense of Adams and Loustaunau (see Definition 4.2.1 and Theorem 4.2.3 in [1]).*

The definition of the set Syz_k depends on the sets G, i. e.. $\mathrm{Syz}_k = \mathrm{Syz}_{G,k}$. The following set will be used in Algorithm 5:

$$\mathrm{Syz}^{(G)} := \mathrm{Syz}_{G,|G|} \tag{1.18}$$

Lemma 1.4.3. *For every element* $\tilde{s} := u_s \cdot s - \sum_{i=1}^{l} a_i \epsilon_i \in S_k$ *we have*

$$\mathrm{LT}(\tilde{s}) = u_s \cdot c_k \cdot \frac{m}{\mathrm{LM}(f_k)} \cdot \epsilon_k$$

with $m \in C_k$ *and* $c \in \mathrm{cSyz}_{k,m}$.

Proof. Let $\tilde{s} = u_s \cdot s - \sum_{i=1}^{l} a_i \epsilon_i$ with $s \in \mathrm{Syz}_k$ and therefore $u_s \cdot \varphi(s) - \mathrm{NF}(\varphi(s) \mid G) = \sum_{i=1}^{l} a_i f_i$ a standard representation. We have $\mathrm{LM}\left(\frac{m}{\mathrm{LM}(f_i)} f_i\right) = \mathrm{LM}\left(\frac{m}{\mathrm{LM}(f_j)} f_j\right)$ for $i, j \in J_{k,m}$ by construction of $J_{k,m}$. Since the leading monomials coincide the component with the highest index will be the leading one by definition of the Schreyer ordering.

Hence $\mathrm{LM}(s) = c_k \cdot \frac{m}{\mathrm{LM}(f_k)}\epsilon_k$ as $k = \max J_{k,m}$ and $c_k \neq 0$ by definition of Syz_k. The leading monomial $\mathrm{LM}\left(\frac{m}{\mathrm{LM}(f_k)} f_k\right) > \mathrm{LM}(\varphi(s))$ since the leading coefficient sum up to zero, i. e. $\sum_{i \in J_{k,m}} c_i \,\mathrm{LC}(f_i) = 0$. Further $\mathrm{LM}(\varphi(s)) \geq \mathrm{LM}(a_i) \cdot \mathrm{LM}(f_i)$ due to the properties of standard representations (see Proposition 1.3.4). Therefore $\mathrm{LM}(\tilde{s}) = u_s \cdot \mathrm{LM}(s) = u_s \cdot c_k \frac{m}{\mathrm{LM}(f_k)}\epsilon_k$. □

Theorem 1.4.4 (Buchberger's criterion and the syzygy theorem)**.** *Let $G = \{f_1, \ldots, f_k\}$ be a generating set of $I \subset \bigoplus_{j=1}^{l} R[\mathbf{x}]_< \cdot \mathbf{e}_j = M$. Further let $S \subset \bigcup_{k=1}^{r} S_k$ and assume*

- $\mathrm{L}(S) = \mathrm{L}\left(\bigcup_{k=1}^{r} S_k\right)$ *and*
- *for each $\tilde{s} \in S$, $\varphi(\tilde{s}) = 0$.*

Then the following hold:

(a) G is a standard basis of I (Buchberger's criterion).

(b) S is a standard basis of $\mathrm{Syz}(I)$ *with respect to the Schreyer ordering. In particular, S generates* $\mathrm{Syz}(I)$*.*

Proof. We prove statements (a) and (b) simultaneously.
Take any $f \in I = \langle f_1, \ldots, f_k \rangle$ and a preimage $g \in F_1$ of f,

$$g = \sum_{i=1}^{k} a_i \epsilon_i, \quad f = \varphi(g) = \sum_{i=1}^{k} a_i f_i.$$

To show (a) we assume $f \neq 0$ and to show (b) $f = 0$. Now consider a polynomial weak normal form h of g with respect to $S \subset F_1$ and the corresponding standard representation of $ug - h$, u a unit in A

$$ug - h = \sum_{\tilde{s} \in S} a_{\tilde{s}} \cdot \tilde{s}.$$

In case of $h = 0$ the substitution morphism φ of g will be zero

$$uf = \varphi(ug) = \sum_{\tilde{s} \in S} a_{\tilde{s}} \cdot \varphi(\tilde{s}) = 0,$$

since $\varphi(\tilde{s}) = 0$ by assumption. In case (a) we had chosen $f \neq 0$ and in case (b) it would be $\mathrm{NF}(g \mid S) = h = 0$. Therefore we may assume $h \neq 0$ and

$$h = h_1\epsilon_1 + \cdots + h_k\epsilon_k,$$

such that $\mathrm{LT}(h_\nu \epsilon_\nu) \notin \mathrm{L}(S)$ or $h_\nu = 0$ for $\nu = 1, \ldots, k$ by properties of weak normal forms on modules (Lemma 1.3.5). Hence we have

$$\mathrm{LT}(h_\nu) \cdot \epsilon_\nu \notin \mathrm{L}(S_\nu) = \mathrm{L}(S) \cap A \cdot \epsilon_\nu \tag{1.19}$$

for all ν with $h_\nu \neq 0$.

Since $\varphi(\tilde{s}) = 0$ by assumption the following equality is given

$$uf = \varphi(ug) = \varphi(h) = \sum_{i=1}^{k} h_i f_i.$$

In case (a) we show, that $\sum_{i=1}^{k} h_i f_i$ is a standard representation of f, i.e. $\mathrm{LM}(f) \geq \mathrm{LM}(h_i)\,\mathrm{LM}(f_i)$ where $h_i \neq 0$. Hence the set G is a standard basis for I by Theorem 1.3.17. In case (b) we see, that $h \neq 0$ give rise to a contradiction and therefore $\mathrm{NF}(g \mid S) = 0$ for all $g \in \mathrm{Syz}(G)$. As before Theorem 1.3.17 states that S is a standard basis of $\mathrm{Syz}(G)$.

Assume $\mathrm{LM}(f) < \mathrm{LM}(h_i)\,\mathrm{LM}(f_i)$ for some i and hence

$$\sum_{i \in J} \mathrm{LT}(h_i)\,\mathrm{LT}(f_i) = 0$$

with $J = \{i \in [k] \mid \mathrm{LM}(h_i)\,\mathrm{LM}(f_i) = \max\}$. Let $k = \max J$ be the index of last element involved in the sum and $m = \mathrm{lcm}(\mathrm{LM}(f_i) \mid i \in J)$ the least common multiple of the leading monomials. Since the leading coefficients sum to zero we have $\sum_{i \in J} \mathrm{LC}(h_i)\,\epsilon_i \in \mathrm{Syz}(\mathrm{LC}(f_i) \mid i \in J_{k,m})$ and may therefore be given by elements of $\mathrm{cSyz}_{k,m}$ as these generates all syzygies of the given form, i. e.

$$\mathrm{LC}(h_k)\,\epsilon_k = \sum_{c \in \mathrm{cSyz}_{k,m}} a_c \cdot c_k \cdot \epsilon_k.$$

Using Lemma 1.4.3 and the definitions of the sets Syz_k and S_r we get

$$\begin{aligned} \mathrm{LT}(h_k)\,\epsilon_k &= m' \cdot \sum_{\tilde{s} \in S_k} a_c \cdot u_s^{-1} \cdot \mathrm{LT}(\tilde{s}) \in \mathrm{L}(S_k) \\ &= m' \cdot \sum_{\tilde{s} \in S_k} a_c \cdot c_k \cdot \frac{m}{\mathrm{LM}(f_k)} \cdot \epsilon_k \in \mathrm{L}(S_k) \end{aligned}$$

for a monomial m'. This is a contradiction to (1.19). □

Remark 1.4.5. *By Corollary 1.4.6 one can implement an algorithm to compute a standard basis over a coefficient ring where linear equations are solvable. The only coefficient ring dependent set is* $\mathrm{cSyz}_{k,m}$. *All others sets may be constructed without specific knowledge about the coefficient ring except that normal forms have to be computable. We will see in Chapter 2 that for a wide range of rings quite a huge simplification is possible. In particular, we can avoid the solution of linear equations in principal ideal rings which makes the algorithm applicable to practical problems.*

The following corollary states requirements on the ground ring sufficient to compute standard bases.

Corollary 1.4.6. *Let R be a commutative noetherian ring with one where linear equations are solvable. Further let $I = \langle f_1, \ldots, f_s \rangle \subset \bigoplus_{i=1}^{m} R[\mathbf{x}]\mathbf{e}_i$ be a submodule and $<$ an arbitrary module monomial ordering. Then a standard basis of I is computable by Algorithm 5.*

Algorithm 5 Computes a standard basis of I

Input: R a ring where linear equations are solvable
Input: M a free $R[\mathbf{x}]_<$-module
Input: $>$ a (module) monomial ordering on M
Input: NF a component-reduced weak normal form on M (e. g. Algorithm 1)
Input: $I = \{f_1, \ldots, f_l\} \subset M \cap \bigoplus_{i=1}^{m} R[\mathbf{x}] \cdot \mathbf{e}_i$ a finite set of module elements
Output: $S = \{s_1, \ldots, s_r\}$ is a standard basis of the submodule generated by I

```
S := I as a ordered set, for the purpose of syzygies
P ⊂ ⋃_{i=1}^{l} Syz^({f_1,...,f_i}) as defined in (1.18) such that the leading ideal of Pis maximal
while P ≠ ∅ do
  choose (g_1, ..., g_r) ∈ P, the cardinality of S is r
  P := P\{(g_1, ..., g_r)}
  h := NF (g_1 s_1 + ··· + g_r s_r | S)
  if h ≠ 0 then
    S := S ∪ {h}
    C ⊂ Syz^(S) with L (C) = L (Syz^(S))
    P := (P × {0}) ∪ {h ∈ C | h_r ≠ 0}
  end if
end while
return S
```

Proof. At first note, that in every run the ideal $\mathrm{L}(S)$ is enlarged if $h \neq 0$. Due to the noetherian property this may only happen on finitely many occasions. Hence after some finite number of iterations h will be equal to zero in every subsequent run. In each run only finitely many elements were added to P given $h \neq 0$. Hence we will reach $P = \emptyset$ after a further finite number of steps.

Let G be the output of algorithm. Then S generated from all used elements $\mathrm{Syz}^{(S)}$ during the algorithm fulfills the properties required by Theorem 1.4.4 and hence G is a standard basis of I

The set P only increases if an element h will be added to S with $\mathrm{LT}(h) \notin \mathrm{L}(S)$. Since the ring R is noetherian so is $R[\mathbf{x}]$ and M. Hence $\mathrm{L}(S)$ can only be enlarged finitely many times. After $\mathrm{L}(S)$ becomes stationary all normal form computations will result in $h = 0$. Therefore $P = \emptyset$ after finitely many loops and the algorithm terminates. □

Chapter 2.

Application to special rings

2.1. General criteria

We use the notations introduced in Section 1.4. At first note that any algorithm based on Theorem 1.4.4 has to prove, that for all elements in $\tilde{s} = s - \sum_{i=1}^{r} a_i \epsilon_i \in S$ the substitution map $\varphi(\tilde{s})$ is zero, i. e. the normal form $\mathrm{NF}(s \mid \cdot)$ of s is zero. Hence, without further information about the coefficient ring R, there are two ways to reduce the complexity of any algorithm based on Theorem 1.4.4.

1. Choose the set S as small as possible.
2. Add as many elements to S for which the normal form computation is easy. At best the normal form is known to be zero without further computation.

Since the equality $\mathrm{L}(S) = \mathrm{L}\left(\bigcup_{k=1}^{r} S_k\right)$ is required in Theorem 1.4.4, this is in fact the limiting condition for 1., hence we have to consider this equality. Lemma 1.4.3 states that the leading terms of elements $\tilde{s} \in S_k$ are of the form

$$\mathrm{LT}(\tilde{s}) = u_s \cdot c_k \cdot \frac{m}{\mathrm{LM}(f_k)} \cdot \epsilon_k.$$

Further we know, that

$$\mathrm{L}(S) = \mathrm{L}\left(\bigcup_{k=1}^{r} S_k\right) \Longleftrightarrow \forall k \,:\, \mathrm{L}(\tilde{s} \in S \mid \mathrm{LComp}(\tilde{s}) = \epsilon_k) = \mathrm{L}(S_k).$$

Hence for any $\tilde{s}_1, \tilde{s}_2 \in S$ the element $\tilde{s}_2$ can be omitted from S if $c_k^{(1)} \,\Big|\, c_k^{(2)}$ and $m^{(1)} \mid m^{(2)}$. An example for 2. is the product criterion given below.

Lemma 2.1.1 (product criterion)**.** *Let $f, g \in R[\mathbf{x}]$ be polynomials with coprime leading monomials (i. e. $\mathrm{lcm}(\mathrm{LM}(f), \mathrm{LM}(g)) = \mathrm{LM}(f) \cdot \mathrm{LM}(f)$) such that $\mathrm{LC}(f), \mathrm{LC}(g)$ are units. For any syzygy of the leading coefficients $(a_f, a_g) \in \mathrm{Syz}(\mathrm{LC}(f), \mathrm{LC}(g))$ with units a_f, a_g the corresponding polynomial $a_f \cdot \mathrm{LM}(g) \cdot f + a_g \cdot \mathrm{LM}(f) \cdot g \in \langle f, g \rangle$ will reduce to zero. In formulas:*

$$\begin{aligned} s &= a_f \cdot \mathrm{LM}(g)\, f + a_g \cdot \mathrm{LM}(f)\, g \\ \mathrm{NF}(s \mid \{f, g\}) &= 0 \end{aligned}$$

Proof. We will show, that

$$1 \cdot s = u_0 \cdot s_0 = hf + kg \tag{2.1}$$

with a unit $u_0 \in R$ where h, k have smaller leading monomials and fewer monomials than g, f, respectively. Further we proof that any such expression ($u_n \cdot s_n \neq 0$) can be rewritten or reduced by G to

$$u_{n+1} \cdot s_{n+1} = h'f + k'g \text{ or } hf + k'g \text{ or } h'f + kg \tag{2.2}$$

such that $h' = \operatorname{tail}(h)$ and $k' = \operatorname{tail}(k)$ and a unit $u_{n+1} \in R[\mathbf{x}]_<$. Since the representation of s_{n+1} fulfills the same properties and polynomials consists of finitely many terms the polynomial s will reduce to zero after finitely many steps as stated. It suffices to show this for an arbitrary normal form since

$$\mathrm{NF}_1(f \mid G) = 0 \Longleftrightarrow \mathrm{NF}_2(f \mid G) = 0 \Longleftrightarrow f \in \langle G \rangle .$$

We will use NF-MGPS (Algorithm 2) and start with computing the polynomial associated to the syzygy (a_f, a_g).

Let $c_f = \operatorname{LC}(f), \mathbf{x}_f = \operatorname{LM}(f)$ and $f' = \operatorname{tail}(f)$ (for g analogously), i. e.

$$\begin{aligned} f &= c_f \cdot \mathbf{x}_f + f' \\ g &= c_g \cdot \mathbf{x}_g + g' \end{aligned}$$

and $(a_f, a_g) \in \operatorname{Syz}(\operatorname{LC}(f), \operatorname{LC}(g))$. Since the leading coefficients and a_f, a_g are units there exists a unit u with $(a_f, a_g) = u \cdot (c_g, -c_f)$. We can express the leading terms of the polynomials by

$$c_f \cdot \mathbf{x}_f = f - f' \text{ and } c_f \cdot \mathbf{x}_g = g - g' \tag{2.3}$$

and hence compute

$$\begin{aligned} s &= u^{-1} a_f \cdot \mathbf{x}_g \cdot f + u^{-1} a_g \cdot \mathbf{x}_f \cdot g \\ &= c_g \cdot \mathbf{x}_g \cdot f - c_f \cdot \mathbf{x}_f \cdot g \\ &= (g - g')f - (f - f')g \\ &= -g'f + f'g = hf + kg. \end{aligned}$$

Note that f', g' are subpolynomials of f, g with fewer monomials and for their leading monomials holds $\operatorname{LM}(f') < \operatorname{LM}(f), \operatorname{LM}(g') < \operatorname{LM}(g)$. Hence a representation of s as stated in (2.1) with $u_0 = 1$is found.

Now let

$$u_n s_n = hf + kg$$

with h, k having smaller leading monomials as g, f and $u_n \in R$ a unit.

At first consider the case where the leading terms sum up to zero, i. e. $\operatorname{LT}(h)\operatorname{LT}(f) + \operatorname{LT}(k)\operatorname{LT}(g) = 0$. Since $\operatorname{lcm}(\operatorname{LM}(f), \operatorname{LM}(g)) = \operatorname{LM}(f) \cdot \operatorname{LM}(g)$ a term t exists, such that

$$t \cdot \operatorname{LT}(g) = \operatorname{LT}(h) \text{ and } -t \cdot \operatorname{LT}(f) = \operatorname{LT}(k) .$$

The inequality of the given leading terms of g an h, i. e. $t\,\mathrm{LM}(g) = \mathrm{LM}(h) < \mathrm{LM}(g)$, yield $t < 1$ by the multiplicative property of the monomial ordering. Hence $u_{n+1} = u_n - t \in R[\mathbf{x}]_<$ is a unit. Now we compute

$$\begin{aligned} u_n s_n &= hf + kg \\ &= t\,\mathrm{LT}(g)\,f + h'f - t\,\mathrm{LT}(f)\,g + k'g \\ &= t(\mathrm{LT}(g)\,f - \mathrm{LT}(f)\,g) + h'f + k'g \\ &= tu^{-1}us_n + h'f + k'g \end{aligned}$$

and hence as stated

$$\underbrace{(u_n - t)}_{u_{n+1}} \cdot \underbrace{s_n}_{s_{n+1}} = h'f + k'g\,.$$

Now assume $\mathrm{LT}(h)\,\mathrm{LT}(f) + \mathrm{LT}(k)\,\mathrm{LT}(g) \neq 0$ and without loss of generality let $\mathrm{LM}(h)\,\mathrm{LM}(f) = \mathrm{LM}(hf + kg)$. Then $u_n \cdot s_n = hf + kg$ will reduce to $s_{n+1} = h'f + kg$ in one step by NF-MGPS. □

2.2. Standard basis over principal ideal rings

Now we show, that in the case of polynomial rings over a principal ideal coefficient ring quite helpful simplifications are possible. We can explicitly construct a special generating set of syzygies in the coefficient rings which will lead to the new definition of s-polynomials. Further in this setting we may directly compute strong standard basis and hence do not need to solve linear equations during (weak) normal form computations.

However, we have to slightly generalized the notion of the s-polynomial as known from the case of coefficient fields due to the presence of zero divisors. Throughout this chapter all coefficient rings are meant to be noetherian principal ideal rings.

Definition 2.2.1. *For any finite number of elements $a_1, a_2, \ldots, a_n \in R$ in the principal ideal ring R we define the*

greatest common divisor *to be an element $g \in R$ that generates the ideal generated by the elements a_i, i. e. $\langle g\rangle = \langle a_1, a_2, \ldots, a_n\rangle$ and*

least common multiple *to be an element $m \in R$ that generated the intersection of all ideal $\langle a_i\rangle$, i. e. $\langle m\rangle = \langle a_1\rangle \cap \langle a_2\rangle \cap \cdots \cap \langle a_n\rangle$.*

Definition 2.2.2. *Let R be a principal ideal ring. The **annihilator** of an element $c \in R$ is the ideal $\mathrm{Ann}(c) = \{c \in R \,|\, a \cdot c = 0\}$ (see Lemma 1.1.6). We will also denote a generator $a \in R$ of the ideal $\mathrm{Ann}(c)$ with $\mathrm{Ann}(c)$ for ease of notation.*

Definition 2.2.3. *Let R be a principal ideal ring and $c_1, c_2 \in R$. The ideal*

$$\langle c_1\rangle : \langle c_2\rangle := \{x \in R \,|\, x \cdot \langle c_2\rangle \subset \langle c_1\rangle\}$$

is generated by an element s of R, i. e. $\langle s_2 \rangle = \langle c_1 \rangle : \langle c_2 \rangle$. Further, since $s_2 \in \langle c_1 \rangle : \langle c_2 \rangle$ there exists an $s_1 \in R$ with $s_1 \cdot c_1 = s_2 \cdot c_2$.

We define

$$\operatorname{smiss}(c_1, c_2) = (s_1, s_2)$$

for an arbitrary choice of a generator s_1 and an element s_2 as above, in particular the relation $s_f c_f - s_g c_g = 0$ holds.

Remark 2.2.4. *If R is a factorial domain or even a factorial ring [29, 2] we have*

$$\operatorname{smiss}(c_1, c_2) = \left(\frac{c_2}{\gcd(c_1, c_2)}, \frac{c_1}{\gcd(c_1, c_2)} \right) = \left(\frac{\operatorname{lcm}(c_1, c_2)}{c_1}, \frac{\operatorname{lcm}(c_1, c_2)}{c_2} \right).$$

Theorem 2.2.5. *Let R be a principal ideal ring and $c = (c_1, c_2, \ldots, c_n) \in R^n$. The syzygies*

$$\operatorname{Syz}(c_1, c_2, \ldots, c_n) = \left\{ x \in R^n \;\middle|\; \sum_{i=1}^{n} x_i \cdot c_i = 0 \right\}$$

of c are generated by

1. $s_{0,i} = (0, \ldots, 0, \underbrace{\operatorname{Ann}(c_i)}_{i\text{-th entry}}, 0, \ldots, 0) \in R^n$ *and*
2. $s_{i,j} = (0, \ldots, \underbrace{s_1^{(i,j)}}_{i\text{-th entry}}, 0, \ldots, 0, \underbrace{s_2^{(i,j)}}_{j\text{-th entry}}, 0, \ldots, 0) \in R^n$

for $0 < i < j \leq n$ and $\operatorname{smiss}(c_i, c_j) = (s_1^{(i,j)}, s_2^{(i,j)})$.

Proof. Let $x \in \operatorname{Syz}(c_1, c_2, \ldots, c_n)$. We will prove the statement by induction n.

For the induction start let $n = 1$, i. e. $x_1 \cdot c_1 = 0$. The element x_1 is in the annihilator ideal $\langle \operatorname{Ann}(c_1) \rangle$ of c_1 and hence $x \in \langle s_{0,1} \rangle$.

Now let $n > 1$. Since $-x_1 \cdot c_1 = \sum_{i=2}^{n} x_i \cdot c_n$ we compute

$$\begin{aligned} -x_n &\in \langle c_1, \ldots, c_{n-1} \rangle : \langle c_n \rangle \\ &= \langle c_1 \rangle : \langle c_n \rangle + \cdots + \langle c_{n-1} \rangle : \langle c_n \rangle \\ &= \left\langle s_2^{(1,n)} \right\rangle + \cdots + \left\langle s_2^{(n-1,n)} \right\rangle \\ &= \sum_{i=1}^{n-1} \alpha_i \cdot s_2^{(i,n)}. \end{aligned}$$

Consider $\tilde{x} = x - \sum_{i=1}^{n-1} \alpha_i \cdot s_{i,n}$. The component $\tilde{x}_n$ is equal to zero and therefore the vector $(\tilde{x_1}, \ldots, \tilde{x}_{n-1})$ is a syzygy in $\operatorname{Syz}(c_1, \ldots, c_{n-1})$. By induction hypothesis the reduced syzygy $\tilde{x} \in \langle s_{0,i}, s_{i,j} \mid 1 \leq i < j < n \rangle$ and hence by definition of $\tilde{x}$

$$x \in \langle s_{0,i}, s_{i,j} \mid 0 < i < j \leq n \rangle.$$

□

Our aim is to use the previous theorem to construct enough elements of the set S of Theorem 1.4.4 to compute a standard basis. In the following we derive these elements and give some criteria allowing to omit some of these elements. In order to give an easy notations we will fix a list $G = \{f_1, f_2, \ldots, f_k\}$ of polynomials and consider the free module $F_1 = \bigoplus_{i=1}^{k} R[\mathbf{x}] \cdot \epsilon_i$ equipped with the Schreyer ordering. Recall that S will be a subset of this free module. We

Definition 2.2.6. *Let $f = c_f \cdot \mathbf{x}_f + \operatorname{tail}(f)$ and $g = c_g \cdot \mathbf{x}_g + \operatorname{tail}(g)$ be elements of $R[\mathbf{x}]\backslash\{0\}$. Further let $\operatorname{smiss}(c_f, c_g) = (s_f, s_g)$. We define the **s-polynomial** of f and g to be*

$$\begin{aligned}\operatorname{spoly}(f,g) :=& s_f \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_f} \cdot f - s_g \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_g} \cdot g \\ =& s_f \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_f} \cdot \operatorname{tail}(f) - s_g \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_g} \cdot \operatorname{tail}(g)\end{aligned}$$

and with the fixed notation the syzygy version

$$\operatorname{spoly}_{\mathrm{Syz}}(f_i, f_j) := s_{f_i} \frac{\operatorname{lcm}\left(\mathbf{x}_{f_i}, \mathbf{x}_{f_j}\right)}{\mathbf{x}_{f_i}} \cdot \epsilon_i - s_{f_j} \frac{\operatorname{lcm}\left(\mathbf{x}_{f_i}, \mathbf{x}_{f_j}\right)}{\mathbf{x}_{f_j}} \cdot \epsilon_j.$$

Remark 2.2.7. *To understand the syzygy version of a s-polynomial consider two polynomials f_i, f_j. Then*

$$\varphi(\operatorname{spoly}_{\mathrm{Syz}}(f_i, f_j)) = \operatorname{spoly}(f_i, f_j).$$

Therefore each syzygy realizes one s-polynomial using the substitution map $\varphi(\epsilon_i) = f_i$.

Remark 2.2.8. *Note that we need* smiss, *i. e. the definition of s-polynomials is not equivalent to*

$$\operatorname{spoly}_r(f,g) = c_g \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_f} f - c_f \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_g} g.$$

For example let $f = 2x - 2y, g = 2y - z$ in $\mathbb{Z}/4[x, y, z]$. Then we get $\operatorname{spoly}_r(f,g) = xz \neq -2y + zx = \operatorname{spoly}(f,g)$. As we have seen, we can loose terms just by multiplying a constant, i.e. if $2x + y \in I$ then we have $2y \in \mathrm{L}(I) \subset I$. Therefore we must bear the consequence of allowing zero divisors by extended the notion of the s-polynomial to also accommodate degree losses due to multiplication with constants.

Definition 2.2.9. *Let $f = c_f \cdot \mathbf{x}_f + \operatorname{tail}(f) \in R[\mathbf{x}]\backslash\{0\}$. We define the **extended s-polynomial** of f to be*

$$\operatorname{spoly}(0, f) = \operatorname{spoly}(f, 0) := \operatorname{Ann}(c_f) \cdot f = \operatorname{Ann}(c_f) \cdot \operatorname{tail}(f) = \varphi(s_{0,f})$$

and with the fixed notation the syzygy version

$$\begin{aligned}\operatorname{spoly}_{\mathrm{Syz}}(0, f_j) = \operatorname{spoly}_{\mathrm{Syz}}(f_j, 0) :=& \operatorname{Ann}\left(c_{f_j}\right) \cdot \epsilon_j \\ \operatorname{spoly}(0, f_j) =& \varphi(\operatorname{spoly}_{\mathrm{Syz}}(0, f_j)).\end{aligned}$$

Algorithm 6 Computes a standard basis of I given a noetherian principal ideal ring

Input:
$f \in R[\mathbf{x}]$ a polynomial, I a finite set of k polynomials,
$>$ a monomial ordering, NF a weak normal form
Output: S is a standard basis of I
$S := \{f_0 = 0, f_1, f_2, \ldots, f_k\}$ as a list
$P := \{(f_i, f_j) \mid f_i, f_j \in S = \{f_0, \ldots, f_k\}, i < j\}$, the pair set
while $P \neq \emptyset$ **do**
choose $(f, g) \in P$
$P := P \backslash \{(f, g)\}$
$h := \mathrm{NF}\left(\mathrm{spoly}\left(f, g\right) \mid S\right)$
if $h \neq 0$ **then**
$P := P \cup \{(f, h) \mid f \in S\}$
$S := S \cup \{h\}$
end if
end while
return S

Corollary 2.2.10. *Algorithm 6 terminates and is correct, i. e. computes a standard basis.*

Proof. Since the syzygies behind the s-polynomials used in the algorithm give rise to the sets S_k of Theorem 1.4.4 this is an easy consequence of the mentioned theorem and Corollary 1.4.6. □

Lemma 2.2.11 (strong product criterion)**.** *Let R be a principal ideal ring and $<$ a global ordering. Further let $f = c_f \cdot \mathbf{x}_f + \mathrm{tail}\left(f\right)$ and $g = c_g \cdot \mathbf{x}_g + \mathrm{tail}\left(g\right)$ be elements of $R[\mathbf{x}]\backslash\{0\}$ with $\mathbf{x}_f$ and $\mathbf{x}_g$ relatively prime, i. e. $\mathrm{lcm}\left(\mathbf{x}_f, \mathbf{x}_g\right) = \mathbf{x}_f \cdot \mathbf{x}_f$. If c_g is an unit and for all terms t_1 of f, t_2 of $\mathrm{tail}\left(f\right)$ and t_3 of $\mathrm{tail}\left(g\right)$*

$$t_2 \cdot c_g \mathbf{x}_g \neq t_3 \cdot t_1$$

holds then

$$\mathrm{NF}\left(\mathrm{spoly}\left(f, g\right) \mid \{f, g, \mathrm{spoly}\left(0, f\right), \mathrm{spoly}\left(0, \mathrm{spoly}\left(0, f\right)\right), \ldots\}\right) = 0.$$

Remark 2.2.12. *The set $\{\mathrm{spoly}\left(0, f\right), \mathrm{spoly}\left(0, \mathrm{spoly}\left(0, f\right)\right), \ldots\}$ is finite as each element has at least one term less as its predecessor in the definition. As a short hand notation let*

$$\begin{aligned}
&\mathrm{spoly}^0(0, f) = f,\\
&\mathrm{spoly}^1(0, f) = \mathrm{spoly}\left(0, f\right),\\
&\mathrm{spoly}^2(0, f) = \mathrm{spoly}\left(0, \mathrm{spoly}\left(0, f\right)\right),\\
&\mathrm{spoly}^3(0, f) = \mathrm{spoly}\left(0, \mathrm{spoly}\left(0, \mathrm{spoly}\left(0, f\right)\right)\right) \text{ and so forth.}
\end{aligned}$$

Proof. Let f, g be as stated, i.e.

$$f = c_f \cdot \mathbf{x}_f + f'$$
$$g = c_g \cdot \mathbf{x}_g + g'$$

where c_g is a unit in R. Since $\langle c_g \rangle = R$ we get $\langle c_g \rangle : \langle c_f \rangle = R : \langle c_f \rangle = R = \langle u \rangle$ for a unit u and hence $\operatorname{smiss}(c_f, c_g) = (u, u \cdot c_f \cdot c_g^{-1})$. We express the leading terms of the polynomials by

$$c_f \mathbf{x}_f = f - f' \text{ and } c_g \mathbf{x}_g = g - g'. \tag{2.4}$$

We start to calculate the normal form of the s-polynomial. Note that, even if there are different normal forms around, it suffices to show this for an arbitrary normal form, due to the fact

$$\mathrm{NF}_1(f \mid G) = 0 \iff \mathrm{NF}_2(f \mid G) = 0 \iff f \in \langle G \rangle_R [\mathbf{x}].$$

Therefore we can choose every reduction step to suit us best, as long as we achieve zero in the end.

$$\begin{aligned} c_g \cdot u^{-1} \operatorname{spoly}(f, g) &= c_g \mathbf{x}_g f - c_f \mathbf{x}_f g && \text{using (2.4)} \\ &= (g - g')f - (f - f')g \\ &= f'g - g'f \end{aligned}$$

Note that f', g' are subpolynomials of f, g with a strictly smaller leading monomial. We will show, that we can reduce one of them without changing this property. But both of them consists only of finitely many terms, therefore after a finite number of steps they will reduce to zero.

At first we must pay tribute to the fact, that $\operatorname{LM}(hk) = \operatorname{LM}(h) \operatorname{LM}(k)$ does not necessarily hold in our case. But to start easy, at first we assume, that $\operatorname{LM}(g'f) = \operatorname{LM}(g') \operatorname{LM}(f)$. Note that, since the leading coefficient c_g of g is a unit, we have $\operatorname{LM}(f'g) = \operatorname{LM}(f') \operatorname{LM}(g)$. Further $\operatorname{LM}(f') \operatorname{LM}(g) = \operatorname{LM}(f) \operatorname{LM}(g')$ cannot hold, since $\operatorname{LM}(g), \operatorname{LM}(f)$ are relatively prime and $\operatorname{LM}(g) \mid \operatorname{LM}(g')$ would imply $\operatorname{LM}(g) \leq_{nat} \operatorname{LM}(g')$ and hence $\operatorname{LM}(g) < \operatorname{LM}(g')$ a contradiction. Therefore we can assume w.l.o.g. $\operatorname{LM}(f'g) < \operatorname{LM}(fg')$. New we can make one reduction step with f and achieve

$$f'g - g'f - \operatorname{LT}(g') f = f'g - (g' - \operatorname{LT}(g')) f.$$

Now consider $\operatorname{LM}(g'f) < \operatorname{LM}(g') \operatorname{LM}(f)$, i. e. $\operatorname{LM}\left(\operatorname{LC}\left(f'_g\right) f\right) < \operatorname{LM}(f)$. Since the conditions on the terms $t_1, t_2, t_3, c_g \mathbf{x}_g$ we have $\operatorname{LM}(g'f) \neq \operatorname{LM}(f'g)$. If $\operatorname{LM}(g'f) < \operatorname{LM}(f'g)$ we can reduce by g as before. Hence assume that $\operatorname{LM}(g'f) > \operatorname{LM}(f'g)$. Now we can reduce by $f_i = \operatorname{spoly}^i(0, f)$ for some i. Let m be the term, such that $mf_i = \operatorname{LT}(g'f)$. Then it holds $mf_i = \operatorname{LT}(g') f$ and therefore we can proceed as above. □

Example 2.2.13. *The polynomials $4x + y$ and $y^2 + 2z \in \mathbb{Z}/8[x, y, t]$ will reduce to zero after the strong product criterion. In contrast $4y + x^3 + 1$ and $x^5 + 2x^2$ will reduce to $2x^2$, which is not reducible by either of the polynomials nor their extended s-polynomials.*

Remark 2.2.14. *The strong product criterion is also true for arbitrary orderings using similar arguments as in Lemma 2.1.1. We will only use the criterion for global orderings in this thesis and hence opted to use the more direct proof.*

Remark 2.2.15. *Even more than in the classical case we see, that the product criterion is a boundary condition. We do not only require, that the monomials are coprime, we further need either that both leading coefficients are units, or one is a unit and an even stronger divisible relation must hold for this monomial. We will do some numerical experiments to evaluate, whether further research in the product criterion is worthwhile. However, one can still divide out common factors of the polynomials as in the classical case to further strengthen the criterion. Whether this results in an if and only if statement was not investigated.*

Now we list a number of criteria which allows to skip certain s-polynomials during the standard basis algorithm and hence speed up the computation by orders of magnitudes. Lets recall that the ring R is a noetherian principal ideal ring.

Corollary 2.2.16 (Product criterion)**.** *With the fixed notation let $f_i, f_j \in G$ with $\operatorname{lcm}\operatorname{LM}(f_i), \operatorname{LM}(f_j) = \operatorname{LM}(f_i) \cdot \operatorname{LM}(f_j)$. Further let $\operatorname{LC}(f_i)$ and $\operatorname{LC}(f_j)$ be a unit, then*

$$\operatorname{NF}(\operatorname{spoly}(f_i, f_j) \mid \{f_i, f_j\}) = 0.$$

Lemma 2.2.17 (Annihilator criterion)**.** *With the fixed notation let $\operatorname{LT}(f_i) = c_i\mathbf{x}^{\alpha_i}$ and $\operatorname{LT}(f_j) = c_j\mathbf{x}^{\alpha_j}$ with $i < j$. If $\operatorname{Ann}(c_j)$ divides s_j with $\langle s_j\rangle = \langle c_i\rangle : \langle c_j\rangle$ (compare to the Definition 2.2.6 of s-polynomials) then*

$$\operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_j)\right) \in \left\langle \operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(0, f_j)\right)\right\rangle.$$

In particular, if $\operatorname{spoly}_{\operatorname{Syz}}(0, f_j) \in S$ then $S\backslash\{\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_j)\}$ is already a standard basis of $\operatorname{Syz}(I)$.

Proof. By definition of the extended s-polynomials $\operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(0, f_j)\right) = \operatorname{Ann}(c_j) \cdot \epsilon_j$ and of the s-polynomials $\operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_j)\right) = s_j \cdot \epsilon_j$. Hence the leading term of the s-polynomial $\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_j)$ is divisible by $\operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(0, f_j)\right)$ and the lemma follows. □

Lemma 2.2.18 (Chain criterion)**.** *With the fixed notation let $\operatorname{LT}(f_i) = c_i\mathbf{x}^{\alpha_i}, \operatorname{LT}(f_j) = c_j\mathbf{x}^{\alpha_j}$ and $\operatorname{LT}(f_l) = c_l\mathbf{x}^{\alpha_l}$ with $i < j < l$. If $c_j\mathbf{x}^{\alpha_j}$ divides $\operatorname{lcm}(c_i\mathbf{x}^{\alpha_i}, c_l\mathbf{x}^{\alpha_l})$ then $\operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_l)\right) \in \left\langle \operatorname{LT}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_j, f_l)\right)\right\rangle$. In particular, if $\operatorname{spoly}_{\operatorname{Syz}}(f_j, f_l) \in S$ then $S\backslash\{\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_l)\}$ is already a standard basis of $\operatorname{Syz}(I)$.*

Proof. The divisibility of $\operatorname{lcm}(c_i\mathbf{x}^{\alpha_i}, c_l\mathbf{x}^{\alpha_l})$ by $c_j\mathbf{x}^{\alpha_j}$ implies

$$\operatorname{lcm}(\mathbf{x}^{\alpha_j}, \mathbf{x}^{\alpha_l}) \mid \operatorname{lcm}(\mathbf{x}^{\alpha_i}, \mathbf{x}^{\alpha_l}).$$

Dividing both sides by $\mathbf{x}^{\alpha_l}$ yields $\operatorname{LM}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_j, f_l)\right) \mid \operatorname{LM}\left(\operatorname{spoly}_{\operatorname{Syz}}(f_i, f_l)\right)$.

Further $c_j \mid \operatorname{lcm}(c_i, c_l)$ implies $\langle c_i\rangle \cap \langle c_l\rangle \subset \langle c_j\rangle$. We compute

$$\begin{aligned}
\left\langle \mathrm{LC}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_i, f_l)\right)\right\rangle &= \langle c_i\rangle : \langle c_l\rangle && \text{by definition}\\
&= \langle c_i\rangle \cap \langle c_l\rangle : \langle c_l\rangle && \text{since } a\cdot\langle c_l\rangle \subset \langle c_l\rangle \text{ for every } a \in R\\
&\subset \langle c_j\rangle : \langle c_l\rangle && \text{as } \langle c_i\rangle \cap \langle c_l\rangle \subset \langle c_j\rangle\\
&= \left\langle \mathrm{LC}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_j, f_l)\right)\right\rangle .
\end{aligned}$$

Together we see $\mathrm{LT}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_j, f_l)\right) \mid \mathrm{LT}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_i, f_l)\right)$ and this proves the statement. □

Now we introduce gcd-polynomials. If these polynomials are added to the critical elements set and if we use an easy normal form, then the resulting standard basis will be strong (see Theorem 2.2.23).

Definition 2.2.19. *Let $f = c_f \cdot \mathbf{x}_f + \mathrm{tail}(f)$ and $g = c_g \cdot \mathbf{x}_g + \mathrm{tail}(g)$ be elements of $R[\mathbf{x}]\backslash\{0\}$ and $\langle h\rangle = \langle c_f, c_g\rangle$. Further let $d_f, d_g \in R$ with $h = d_f \cdot c_f + d_g \cdot c_g$. We define the* gcd-***polynomial*** *of f and g to be*

$$\begin{aligned}
\text{gcd-poly}(f,g) :=& d_f \cdot \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_f} \cdot f + d_g \cdot \frac{\operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g)}{\mathbf{x}_g} \cdot g \text{ and}\\
\mathrm{gmiss}(f,g) :=& (d_f, d_g)
\end{aligned}$$

and with the fixed notation

$$\begin{aligned}
\text{gcd-poly}_{\mathrm{Syz}}(f_i, f_j) :=& d_{f_i} \cdot \frac{\operatorname{lcm}\left(\mathbf{x}_{f_i}, \mathbf{x}_{f_j}\right)}{\mathbf{x}_{f_i}} \cdot \epsilon_i + d_{f_j} \cdot \frac{\operatorname{lcm}\left(\mathbf{x}_{f_i}, \mathbf{x}_{f_j}\right)}{\mathbf{x}_{f_j}} \cdot \epsilon_j\\
\text{gcd-poly}(f_i, f_j) =& \varphi(\text{gcd-poly}_{\mathrm{Syz}}(f_i, f_j)).
\end{aligned}$$

Remark 2.2.20. *If already one leading coefficient generates the ideal $\langle c_f, c_g\rangle = \langle c_f\rangle$ or $\langle c_f, c_g\rangle = \langle c_f\rangle$ then the leading monomial $\mathrm{LT}(f)$ of f divides the* gcd*-polynomial* $\text{gcd-poly}(f,g)$ *or* $\mathrm{LT}(g) \mid \text{gcd-poly}(f,g)$ *respectively. Hence the* gcd*-polynomial is not required to be considered. It is important to notice, that the* gcd*-polynomials can not be omitted by the same reasoning as for the s-polynomials. The* gcd*-polynomials shows missing elements to allow that each reduction step is only using on element of the basis while the s-polynomials are concerned with the size of the leading ideal. The* gcd*-polynomials will never increase the size of the leading ideal as there leading term can be constructed from the leading terms of polynomials f_i and f_j. Nevertheless one may use the* gcd*-polynomials to omit s-polynomials, as the following criteria demonstrate.*

Remark 2.2.21. *If R is a factorial domain or even a factorial ring [29, 2] we have*

$$\mathrm{LT}(\text{gcd-poly}(f,g)) = \gcd(c_f, c_g) \cdot \operatorname{lcm}(\mathbf{x}_f, \mathbf{x}_g).$$

Lemma 2.2.22 (gcd criterion)**.** *With the fixed notation let $\mathrm{LT}(f_i) = c_i\mathbf{x}^{\alpha_i}, \mathrm{LT}(f_j) = c_j\mathbf{x}^{\alpha_j}$ and $\mathrm{LT}(f_l) = c_l\mathbf{x}^{\alpha_l}$ with $i, j < l$. If $\gcd(c_j, c_l) \in \langle c_i\rangle : \langle c_l\rangle$ then also the leading term $\mathrm{LT}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_i, f_l)\right) \in \left\langle \mathrm{LT}\left(\text{gcd-poly}_{\mathrm{Syz}}(f_j, f_l)\right)\right\rangle$. In particular, if the* gcd*-polynomial* $\text{gcd-poly}_{\mathrm{Syz}}(f_j, f_l) \in S$ *then $S\backslash\{\mathrm{spoly}_{\mathrm{Syz}}(f_i, f_l)\}$ is already a standard basis of* $\mathrm{Syz}(I)$.

Proof. By definition $\left\langle \mathrm{LT}\left(\text{gcd-poly}_{\mathrm{Syz}}(f_j, f_l)\right)\right\rangle = \langle c_j, c_l\rangle = \langle \gcd(c_j, c_l)\rangle$ and by assumption $\langle \gcd(c_j, c_l)\rangle \subset \langle c_i\rangle : \langle c_l\rangle = \mathrm{LC}\left(\mathrm{spoly}_{\mathrm{Syz}}(f_i, f_l)\right)$. □

Algorithm 7 Computes a strong standard basis of I given a noetherian principal ideal ring

Input:
$f \in R[\mathbf{x}]$ a polynomial, I a finite set of polynomials,
$>$ a monomial ordering, NF a weak normal form for strong standard basis (see Algorithm 3)

Output: S is a standard basis of I

```
S := I = {f_0 = 0, f_1, ..., f_k} as a list
P ≙ the set of critical elements
P := {spoly_Syz(f_i, f_j) | 0 ≤ i < j ≤ k}
P := P ∪ {gcd-poly_Syz(f_i, f_j) | 1 ≤ i < j ≤ k}
while P ≠ ∅ do
  choose p ∈ P
  P := P\{p}
  h := NF(Σ_{i=1}^{k} p_i f_i | S)
  if h ≠ 0 then
    f_{k+1} := h
    P := P × {0} ∪ {spoly_Syz(f_i, f_{k+1}), | f_i ∈ S}
    P := P ∪ {gcd-poly_Syz(f_i, f_{k+1}), | f_i ∈ S}
    Apply all criteria to P to remove unnecessary elements
    S := S ∪ {f_{k+1}}
    k := k + 1
  end if
end while
return S
```

Theorem 2.2.23. *Let R be a principal ideal ring and G a standard basis. Then G is a strong standard basis if and only if for all $f_1, f_2 \in G$ there exists a $g \in G$ with*

$$\mathrm{LT}\left(\text{gcd-poly}\left(f_1, f_2\right)\right) \in \langle \mathrm{LT}\left(g\right)\rangle .$$

Proof. Assume G be a strong standard basis. Since $\text{gcd-poly}\left(f, g\right) \in \langle G\rangle$ Definition 1.3.6 states that an element $h \in G$ as required exist.

Assume the condition on the leading terms of the gcd-polynomials is fulfilled. Let $h \in \langle G\rangle$. Since G is a standard basis there are terms t_i such that $\mathrm{LT}\left(h\right) = \sum_{i=0}^{n} t_i \cdot \mathrm{LT}\left(f_i\right)$ with $f_i \in G$ and $\mathrm{LM}\left(t_i\right)\mathrm{LM}\left(f_i\right) = \mathrm{LM}\left(h\right)$. We will prove by induction on n that an element $g \in G$ with $\mathrm{LT}\left(g\right) | \mathrm{LT}\left(h\right)$ exists. For $n = 1$ there is nothing to show.

Now let $n > 1$.

$$\begin{aligned}
\mathrm{LT}(h) &= \sum_{i=0}^{n} t_i \cdot \mathrm{LT}(f_i) \\
&= t_1 \cdot \mathrm{LT}(f_1) + t_2 \cdot \mathrm{LT}(f_2) + \sum_{i=2}^{n} t_i \cdot \mathrm{LT}(f_i) \\
&= t \cdot \mathrm{LT}(\text{gcd-poly}(f_1, f_2)) + \sum_{i=2}^{n} t_i \cdot \mathrm{LT}(f_i) \text{ by definition of gcd-poly} \\
&= t \cdot s \cdot \mathrm{LT}(g) + \sum_{i=2}^{n} t_i \cdot \mathrm{LT}(f_i) \text{ by requirement of Theorem}
\end{aligned}$$

Hence the leading term of h is also the sum of $n-1$ leading terms and hence by induction hypothesis there exists a $g \in G$ with $\mathrm{LT}(g) \mid \mathrm{LT}(h)$. □

Corollary 2.2.24. *Algorithm 7 terminates and computes a strong standard basis of I.*

Proof. Termination follows from Corollary 2.2.10. Same corollary and Theorem 2.2.23 yields that G is a strong standard basis. □

Remark 2.2.25. *The delayed computation of the critical element allows us to make use of the criteria developed above in an efficient way. Further the delayed computation saves memory and computation time.*

2.2.1. Lifting standard bases

If R is a principal ideal ring, then R is isomorphic to a finite product [91] of principal ideal domains and finite-chain rings. The set of ideals in a finite-chain rings can be ordered to a strictly increasing chain $\{0\} \subset I_1 = \langle \eta \rangle \subset I_2 \subset \cdots \subset R$. As R is noetherian the increasing chain is finite and hence every element in a finite-chain ring may be written as $u \cdot \eta^i$ for a unit u and a natural number i. The special structure speeds standard basis computation up.

Further we can compute Gröbner basis in polynomials rings over the factors and lift them to $R[\mathbf{x}]$. This is described in the work of G. Norton and A. Salagean [58]. Randomized benchmarks showed that computation in the factors and lifting afterwards is faster and more space efficient as computations in the product. A straight-forward lifting implementation as a SINGULAR library can be found at `http://www.mathematik.uni-kl.de/~wienand/liftgb.tar.bz2` (only functional with dedicated SINGULAR versions).

Chapter 3.

Implementation in SINGULAR

In the following chapter we provide a brief overview of the SINGULAR architecture. One of the main features is a highly efficient[1] implementation of the Buchberger algorithm and the Mora, Greuel, Pfister, Schönemann (MGPS) extensions thereof for arbitrary orderings and many coefficient fields [33, 36, 23].

Next we describe the additions and changes necessary to allow computations in the presence of zero divisors (in principal ideal rings). There are two major differences: multiplication of the terms may lead to zero and more critical elements have to be considered.

At last we give some comparisons regarding memory consumption and runtime to the computer algebra system Magma. Magma is the only other system able to compute standard bases in the rings $\mathbb{Z}$ and $\mathbb{Z}/m$ (Magma only for global orderings) we are aware of.

3.1. Singular Architecture (for straight standard basis computations)

The computer algebra system SINGULAR has a several decades long history of active development reaching back to 1984. Hence we will not try summarize or display the overall architecture of the system but just focus on the parts vital for the standard basis computation based on the Buchberger algorithm. The Buchberger algorithm can be described as a completion algorithm in the following sense.

In each round it will take one critical element which may show that the result is not complete and verifies if the current result is sufficient or need to be enlarged. If the result needs to be enlarged further critical elements are created which have to verified on a following round.

In order to execute any computations we need to be able to do basic operations like addition, multiplication, testing for divisibility in the coefficients and the resulting polynomial ring or free module thereof. A defined set of operations is implemented in a separate source code module for each ring. Hence we have implemented the corresponding functions for $\mathbb{Z}$, $\mathbb{Z}/m$ and an optimized implementation for $\mathbb{Z}/2^n$ where n is smaller

[1] Efficient is meant in a practical sense. From the theoretical point of view it is known that GB computations have a (double) exponential, hence non-efficient, complexity

than the machine word bit-size. Further the interface describing the required functions per coefficient ring had to be extended since some functions are only required for rings and not for the already implemented fields, e. g. tests for divisibility or units, greatest common divisors, and so on.

Since operations in polynomial rings and free modules are combinatorial operations of elements in the corresponding coefficient rings they may be abstracted and used for different coefficient rings. The fact that the multiplication of non-zero elements in a field will never result in zero allows quite some optimization in the abstracted algorithms for polynomials and modules based on coefficient fields. Therefore we had to check the implementation of polynomial and module arithmetics in SINGULAR for such optimization and adapted the algorithms with minimal impact accordingly.

An important operation is also the comparison of monomials with respect to the given monomial ordering. Here we use the existing SINGULAR functions.

Of course all new rings and methods need to be exposed to the SINGULAR interpreter in order to be used. All changes described above are the basis to modify the Buchberger implementation of SINGULAR for coefficient rings and may be reviewed in the SINGULAR source code repository at `http://www.singular.uni-kl.de/`.

An abstracted flow chart of the SINGULAR implementation of the Buchberger algorithm is displayed in Figure 3.1. The different blocks and variables of the chart are explained below:

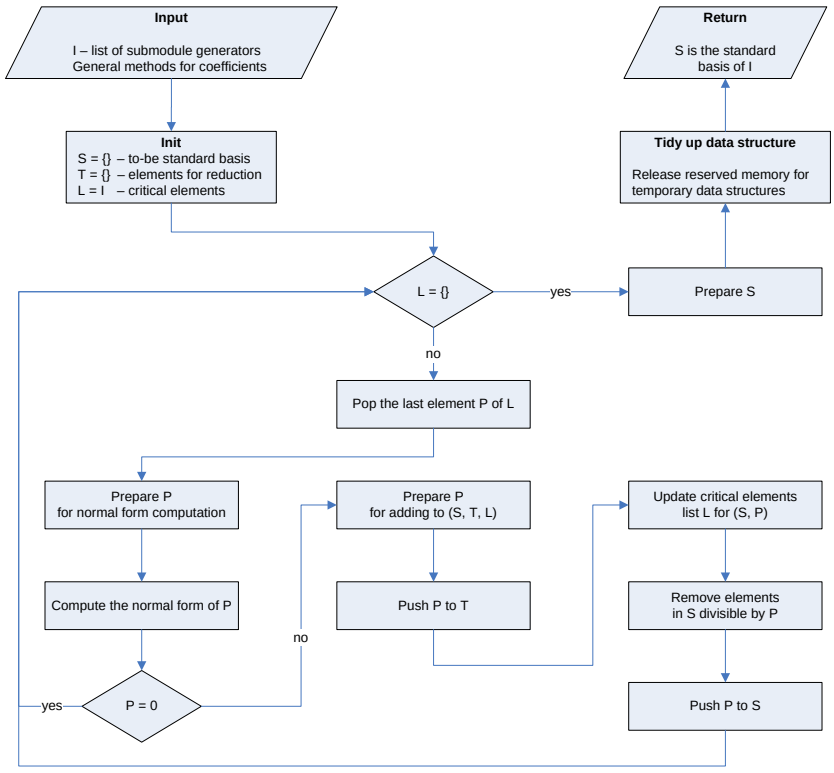


Figure 3.1.: Flow diagram for central Buchberger loop (BBA) in SINGULAR

set I The set contains the generators of the submodule for which the standard basis will be computed.

list S This list contains the current best bet of the algorithm for the standard basis of the submodule generated by I. Initially it will be empty.

list T This list contains all elements of S and further elements which are known to be in the submodule generated by S and can therefore be used for reduction in the normal form computation. Initially it will be empty.

list L This list contains all **critical elements** at the current stage. If all critical elements reduce to zero given S than S is a standard basis of the submodule generated by I. Elements of this list can be thought of as polynomials. But during the algorithm these elements contains a lot more data in order to allow iterative applications of the chain criterion and to delay s-polynomial computations to the latest possible moment, i. e. just before the normal form of the element is computed. Initially it will just contain all elements of I.

3.2. Overview of the SINGULAR's BBA and extensions of coefficient rings

We will give a short description of each process block in the flow diagram Figure 3.1 and describe the changes and extension necessary for standard basis computation with coefficient rings.

Init As already stated in the previous section the lists S, T are initialized as empty lists and L contains just the generators from the set I.

Pop the last element of L The list L is sorted by a definable key to map different selection strategy for the next element by just changing the sorting algorithm. One may consider the length of the polynomials, the ordering of the leading terms, and so on.

Prepare P **for normal form computation** Every element in the list L will be polynomial in the end. But for optimization the actual computation of this polynomial, e. g. the s-polynomial, may be deferred as long as possible. In this process step the polynomial is computed since it is required for the next step.

Compute the normal form of P Here the normal form is computed dependent on the chosen monomial ordering and further parameters by different algorithms, e. g. NF-MGPS. The normal form will be computed against the polynomials of the list T. If the coefficients are not within a field divisibility tests have to be done for them also. Due to further elements which are added to the list L we do not need to solve linear equations in the normal form computation for coefficients in principal ideal rings.

Prepare P for adding to (S, T, L) Dependent on the chosen computation strategy also the tail of the element P may be reduced or other normalizations may be computed, e. g. norming the leading coefficient to one (if possible) or just removing easy common factors of the coefficients.

Push P to T The list T will contain every element which is or was present in the list S during the algorithm. This list is only used for normal form computations.

Update critical elements list L for S, P In this step the following new critical elements are created:

- For each element of S the s-polynomial with P.
- The extended s-polynomial if the $\mathrm{LC}(P)$ of P is a zero divisor.
- For each element of S the gcd-polynomials with P if necessary as described by Theorem 2.2.23.

The last two types of critical elements are only required for coefficient rings. The second is required in order to guarantee that the result is a standard basis, while the third allows us to use an easy normal form algorithm without using linear algebra algorithms in the coefficient ring. Note that this is only possible if the coefficients form a principal ideal ring. Further this process step applies all possible instances of the product (see Lemma 2.1.1) and chain criterion (see Lemma 2.2.18). If the chain criterion allows a choice of elements to be omitted the algorithm tries to omit the most expensive to compute ones. In the best case one the choices is already known to be zero by the product criterion and therefore both choices can be omitted.

Currently the annihilator and gcd criterion (see Lemma 2.2.17 and Lemma 2.2.22) are not implemented in SINGULAR.

Remove elements in S divisible by P Due to the Definition 1.3.6 of standard bases and Lemma 1.3.9 only the size of the submodule $\mathrm{L}(S)$ is important. If for an element $s \in S$ the leading term of P divides s we have $\mathrm{L}(S \backslash \{s\}) \subset \mathrm{L}(S \cup \{P\})$ and hence s can be omitted from S. The fewer elements the list S consists of the fewer elements we need to consider while creating critical elements. Since we added further critical elements to compute a strong standard basis on the fly we cannot remove arbitrary elements s from S such that $\mathrm{L}(S) = \mathrm{L}(S \backslash \{s\})$. Due to the strong property we can only omit elements $s \in S$ for which an element $s' \in S$ with $\mathrm{LT}(s') \mid \mathrm{LT}(s)$ exists.

Push P to S Housekeeping has to be done. The element P was not reduced to zero by a normal form computation but has to since P is an element of the submodule generated by I. In order to guarantee that a normal form computation of our resulting standard basis will reduce P to zero we add it to the standard basis.

Prepare S We may choose to interreduce S, i. e. reduce every element of S by all other elements of S for example.

Tidy up data structures This process step just release all allocated memory areas and temporary data structures allocated during the algorithm.

3.3. Benchmarks

In Section 2.2 we explained, how an efficient algorithm for standard basis computation for principal ideal rings can be instantiated. We implemented the algorithm in the kernel of the computer algebra system SINGULAR [23] for the coefficient rings $\mathbb{Z}$ and $\mathbb{Z}/m$. The performance is compared to Magma using random ideals (for examples derived from our main application, see Table 5.1). Magma is the only other system we found to be capable of computing Gröbner bases in the given rings. For local or mixed orderings we are not aware of a second system which can compute standard bases. In the following tables we present only a few concrete runtimes, but they give an overall impression of the data. The input data may be downloaded from `http://www.mathematik.uni-kl.de/~wienand/phd_benchmarks.tar.bz2`.

All benchmarks were done with SINGULAR 3-1-2 and Magma V2.16-4 on a server running Gentoo (Linux 2.6.32) with 16 GB of memory and an AMD Dual Opteron 2.2 GHz processor. A timeout limit of 1 hour and a memory limit of roughly 12 GB were used.

The new implementation of the MGPS-algorithm for principal ideal rings in SINGULAR is by orders of magnitudes faster than Magma. In most cases where Magma will not terminate within one hour, SINGULAR only requires about a couple of seconds.

The tables are organized as follows:

Columns	Legende
#vars.	number of variable in the polynomial ring
#polys.	number of polynomials in the input generating set
maxdeg	maximal total degree of a monomial in the instance
$\frac{\#\text{mons.}}{\#\text{polys.}}$	combined density of the polynomials, i. e. the number of monomials in all polynomials divided by the number of polynomials in the input generating set
#GB	number of elements in the Gröbner basis (the number was in every case the same for SINGULAR and Magma, if both systems terminated within the given restrictions)

The remaining columns show the CPU times and memory required by the particular tool to compute a Gröbner basis of the input data. In case the memory limit or timeout limit was reached this is indicated in the respective columns.

#vars.	#polys.	maxdeg	$\frac{\#\text{mons.}}{\#\text{polys.}}$	#GB	SINGULAR		Magma	
2	5	15	69.2	3	0.40 s	4.11 MB	68.16 s	13.57 MB
3	3	10	6.7	254	8.50 s	17.23 MB	1287.80 s	19.60 MB
3	3	15	7.4	599	204.82 s	146.98 MB	>1 h	
4	4	10	2.8	120	0.04 s	0.87 MB	10.68 s	9.52 MB
4	4	10	3.0	361	20.36 s	32.24 MB	>1 h	
5	5	10	2.4	584	0.15 s	1.09 MB	455.35 s	30.07 MB
5	5	10	2.8	1043	1.11 s	2.34 MB	>1 h	
7	5	10	2.0	614	0.14 s	1.14 MB	40.06 s	35.35 MB
7	5	10	2.2	2547	2.23 s	3.03 MB	>1 h	
10	10	4	1.9	436	0.11 s	1.09 MB	92.45 s	16.75 MB
10	10	4	3.0	11734	963.39 s	341.70 MB	>1 h	
12	10	3	2.3	5536	18.40 s	16.75 MB	>1 h	
12	10	3	3.0	1940	3.69 s	13.12 MB	>1 h	

Table 3.1.: Randomly generated examples in $\mathbb{Z}_{2^{10}}$ with degree reverse lexicographical ordering.

#vars.	#polys.	maxdeg	$\frac{\#\text{mons.}}{\#\text{polys.}}$	#GB	SINGULAR		Magma	
10	10	4	1.2	30	0.0 s	0 MB	0.0 s	9 MB
10	10	4	1.4	156	0.1 s	1 MB	43.6 s	26 MB
10	10	4	1.4	130	0.0 s	0 MB	0.7 s	11 MB
10	10	4	1.4	105	0.0 s	0 MB	0.2 s	11 MB
10	10	4	1.4	29	0.0 s	0 MB	0.0 s	9 MB
10	10	4	1.6	386	3.3 s	4 MB	>1 h	
10	10	4	1.6	73	0.0 s	0 MB	0.1 s	9 MB
10	10	4	1.7	297	1.4 s	2 MB	>1 h	
10	10	4	1.7	612	1313.2 s	137 MB	>1 h	
10	10	4	1.8	813	487.3 s	55 MB	>1 h	

Table 3.2.: Randomly generated examples in $\mathbb{Z}/m$ with $m = 193697325 = 3^4 \cdot 5^2 \cdot 41 \cdot 2333$ with lexicographical ordering.

#vars.	#polys.	maxdeg	$\frac{\#\text{mons.}}{\#\text{polys.}}$	#GB	SINGULAR		Magma	
10	10	4	1.2	24	0.0 s	0 MB	0.0 s	8 MB
10	10	4	1.4	121	0.1 s	0 MB	4.1 s	16 MB
10	10	4	1.4	133	0.1 s	0 MB	4.6 s	14 MB
10	10	4	1.4	94	0.0 s	0 MB	0.1 s	9 MB
10	10	4	1.4	28	0.0 s	0 MB	0.0 s	9 MB
10	10	4	1.6	316	0.6 s	2 MB	>1 h	
10	10	4	1.6	82	0.0 s	0 MB	0.2 s	9 MB
10	10	4	1.7	328	0.7 s	2 MB	631.5 s	87 MB
10	10	4	1.7	445	17.4 s	10 MB	>1 h	
10	10	4	1.8	794	131.7 s	20 MB	>1 h	

Table 3.3.: Same instances as in Table 3.2 but with degree reverse lexicographical ordering.

#vars.	#polys.	maxdeg	$\frac{\#\text{mons.}}{\#\text{polys.}}$	#GB	Singular		Magma	
4	4	10	1.75	51	0.1 s	0 MB	1.0 s	9 MB
4	4	10	1.75	133	2.2 s	4 MB	1387.9 s	36 MB
4	4	10	2	79	1991.3 s	1801 MB	>1 h	
5	4	7	2	194	53.6 s	32 MB	>1 h	
5	4	7	2	258	116.7 s	49 MB	1353.6 s	71 MB
8	4	4	2	122	0.7 s	4 MB	44.3 s	19 MB
8	4	4	2	67	0.1 s	1 MB	6.0 s	14 MB
8	4	4	2.25	169	15.4 s	69 MB	>1 h	
8	4	4	2.25	193	3.0 s	8 MB	343.1 s	36 MB
8	4	4	2.25	76	0.9 s	9 MB	590.6 s	52 MB
8	4	4	2.25	55	0.0 s	0 MB	1.7 s	11 MB
8	4	4	2.25	131	0.8 s	4 MB	2067.4 s	122 MB
8	4	4	2.25	70	0.2 s	1 MB	24.2 s	18 MB
8	4	4	2.5	207	5.0 s	39 MB	>1 h	
8	4	4	2.5	377	1644.4 s	2986 MB	>1 h	
8	4	4	2.5	161	4.8 s	45 MB	>1 h	
8	4	4	2.5	225	142.7 s	137 MB	>1 h	
8	4	4	2.5	67	0.2 s	1 MB	190.4 s	31 MB
8	4	4	2.75	295	16.5 s	56 MB	>1 h	
8	4	4	2.75	413	219.0 s	136 MB	>1 h	
8	4	4	3	590	987.2 s	567 MB	>1 h	
8	4	4	3	117	1.5 s	18 MB	>1 h	
8	4	4	3	131	5.1 s	16 MB	>1 h	
8	4	4	3	109	8.7 s	27 MB	>1 h	
8	4	4	3.25	97	20.9 s	48 MB	>1 h	
8	4	4	3.75	188	178.2 s	145 MB	>1 h	

Table 3.4.: Randomly generated examples in $\mathbb{Z}$ with lexicographical ordering.

Chapter 4.

Vanishing polynomials

We construct an explicit minimal strong Gröbner basis of the ideal of vanishing polynomials in the polynomial ring over $\mathbb{Z}/m$ for $m \geq 2$. The proof is done in a purely combinatorial way. It is a remarkable fact that the constructed Gröbner basis is independent of the monomial ordering and that the set of leading terms of the constructed Gröbner basis is unique, up to multiplication by units. We also present a fast algorithm to compute reduced normal forms, and furthermore, we give a recursive algorithm for building a Gröbner basis in $\mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ along the prime factorization of m. The obtained results are not only of mathematical interest but have immediate applications in formal verification of data paths for microelectronic systems-on-chip designs as described in Chapter 5.

This chapter is published in the Journal of Symbolic Computation with title "The Groebner basis of the ideal of vanishing polynomials" [37] together with Gert-Martin Greuel und Frank Seelisch.

4.1. Introduction

Although the basic properties of Gröbner bases in polynomial rings over a ring R are well-known (e. g. see [1]), they have not been studied very much, mainly because they were considered as academic, in contrast to the case where the ground ring R is a field. Recently however, Gröbner basis techniques in polynomial rings over $R = \mathbb{Z}/m$ (in particular $\mathbb{Z}/2^k$) have attracted some attention due to their potential applications to proving correctness of data paths in system-on-chip designs (see Chapter 5).

When the underlying ring R has only finitely many elements, then there exist polynomials in $R[x_1, x_2, \ldots, x_n]$ which evaluate to zero for all $(a_1, a_2, \ldots, a_n) \in R^n$, called vanishing polynomials. Thus, any polynomial function $\widetilde{f} : R^n \to R$ given by an arbitrary element $f \in R[x_1, x_2, \ldots, x_n]$, will have many alternative representations in $R[x_1, x_2, \ldots, x_n]$, as $\widetilde{f} = \widetilde{f + g}$, for all g that constantly vanish on R^n. All vanishing polynomials constitute an ideal I_0. If R is field the ideal I_0 is well known and is generated by $\{x_1^p - 1, x_1^p - 1, \ldots, x_n^p - 1\}$ where p is the characteristic of R. In this case algebraic computation concerning functions over the field can be done either by adding the generators or by using correspondingly adapted data structure (as is done in [8] for $\mathbb{Z}/2$).

In the applications mentioned above, not the polynomials but only the polynomial functions are of interest. Thus, if we want to apply algebraic methods we need to be able to efficiently compute normal forms of polynomials with respect to a Gröbner basis of I_0. In this chapter, we set the theoretical ground and provide fast algorithms for doing these computations.

From a mathematical point of view, $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ has some interesting properties. In this chapter, we will give an explicit minimal strong Gröbner basis G_m for I_0. As will turn out, G_m is a Gröbner basis with respect to *every* global monomial ordering. Moreover, we will show for any alternative minimal strong Gröbner basis G of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ that the sets of leading terms of G_m and G are the same up to multiplication by units. This is remarkable, since the ring $\mathbb{Z}/m$ has zero divisors. In general, the leading terms of two minimal strong Gröbner bases of an ideal $I \subset R[x_1, x_2, \ldots, x_n]$ need not be related by a unit but only by some element of R. We will prove both properties and show also that in general all minimal strong Gröbner bases of an arbitrary ideal $I \subset R[x_1, x_2, \ldots, x_n]$ have the same number of elements.

From a practical point of view, as mentioned above, engineering tasks involving the computation of Gröbner bases over finite rings will often need to deal with vanishing polynomials. This is due to the fact that normally the elements of a Gröbner basis G will be used to decide the consistency of a mathematical model. And typically, such a check involves the question whether the set of zeros of all polynomials $f \in G$ coincides with the set of all feasible input-output vectors of the modeled artifact; see also Chapter 5. Our interest was specifically spurred by a cooperation with the local Electronic Design Automation Group in which we use Gröbner bases to formally verify chip designs. More precisely, a given verification task is translated into a polynomial ideal in $\mathbb{Z}/2^k$, where typically $k = 32$ or $k = 64$; see Chapter 5. For the special case of polynomial datapath verification we also refer to [89] in which it was shown that the Gröbner basis approach proves tractable for industrial applications where standard property checking techniques failed.

This chapter is organized as follows. Section 2 briefly recalls the basic concepts from the theory of polynomial rings and Gröbner bases such that this chapter may be read by its own. Section 3 starts by presenting canonical members of the ideal of vanishing polynomials $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$. Next we show that the leading term of any given vanishing polynomial is divisible by the leading term of an appropriate canonical member. This relation enables us to finally construct an explicit minimal strong Gröbner basis G_m of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$. We also show that the size of G_m is polynomial of the degree k in the number of variables n, when we are in the practically relevant case $m = 2^k$.

The theoretical results are followed by algorithms for computing reduced normal forms with respect to the constructed basis, and for recursively computing a Gröbner basis of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ along the prime factorization of m. The normal form algorithm has been implemented in the computer algebra system SINGULAR [36] and successfully applied, [89].

4.2. Preliminaries

In this section we recall all definition necessary for this chapter already given in Chapter 1 and define the term "vanishing polynomial".

Let R be a commutative, noetherian ring with 1, and $R[\mathbf{x}] := R[x_1, x_2, \ldots, x_n]$ a multivariate polynomial ring over R, where $n \geq 1$. For any multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$, a product of variables $\mathbf{x}^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is called a monomial, and a product $a \cdot \mathbf{x}^\alpha$ with $a \in R$ is called a term.

Given two multi-indices $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n)$, we define $\alpha \pm \beta := (\alpha_1 \pm \beta_1, \ldots, \alpha_n \pm \beta_n)$. We may compare α and β according to the predicate $\alpha \preceq \beta :\Leftrightarrow \forall i \in \{1, \ldots, n\} \,:\, \alpha_i \leq \beta_i$, and similarly $\alpha \prec \beta :\Leftrightarrow \alpha \preceq \beta \wedge \alpha \neq \beta$. For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0, 1, 2, \ldots\}^n$, we write $\alpha! := \alpha_1! \cdots \alpha_n!$, and $|\alpha| := \alpha_1 + \ldots + \alpha_n$.

Moreover, we require the polynomial ring $R[\mathbf{x}]$ to be equipped with a global monomial ordering $<$, i. e., $<$ is a well-order on the set of monomials and satisfies $\mathbf{x}^\alpha > \mathbf{x}^\beta \Rightarrow \mathbf{x}^{\alpha+\gamma} > \mathbf{x}^{\beta+\gamma}$ for all $\alpha, \beta, \gamma \in \{0, 1, 2, \ldots\}^n$. Then $<$ refines the partial order $\prec$.

Since we are going to work with divisibility in $\mathbb{Z}/m[x_1, x_2, \ldots, x_n]$, we need to distinguish between divisibility in $\mathbb{Z}/m$ and in $\mathbb{Z}$. We set $a\big|_{\mathbb{Z}} b :\Leftrightarrow \exists\, k \in \mathbb{Z} \,:\, b = a \cdot k$ and $a\big|_m b :\Leftrightarrow \exists\, k \in \mathbb{Z} \,:\, m\big|_{\mathbb{Z}}(b - a \cdot k)$, that is, b and $a \cdot k$ represent the same residue class in $\mathbb{Z}/m$. For two monomials $a\mathbf{x}^\alpha, b\mathbf{x}^\beta$, we say that $a\mathbf{x}^\alpha$ divides $b\mathbf{x}^\beta$, if $a\big|_m b \wedge \alpha \preceq \beta$. We then write $a\mathbf{x}^\alpha | b\mathbf{x}^\beta$, using the ordinary symbol.

Let $f = a_0 \cdot \mathbf{x}^{\alpha^{(0)}} + \cdots + a_k \cdot \mathbf{x}^{\alpha^{(k)}}$ be a polynomial in $R[x_1, x_2, \ldots, x_n]$ with $a_i \neq 0$ for $0 \leq i \leq k$, and $x^{\alpha^{(0)}} > x^{\alpha^{(1)}} > \cdots > x^{\alpha^{(k)}}$. We use the following notation:

$\deg(f) = \max\{\lvert\alpha^{(i)}\rvert \mid 0 \leq i \leq k\}$	total degree of f,
$\mathrm{LT}(f) = a_0 \cdot \mathbf{x}^{\alpha^{(0)}}$	leading term of f,
$\mathrm{LM}(f) = \mathbf{x}^{\alpha^{(0)}}$	leading monomial of f,
$\mathrm{LC}(f) = a_0$	leading coefficient of f,
$\mathrm{L}(A) = \langle \mathrm{LT}(f) \mid f \in A \rangle_{R[x_1,x_2,\ldots,x_n]}$	leading ideal of A,
	for $A \subset R[x_1, x_2, \ldots, x_n]$, $A \neq \emptyset$.

For an ideal $I \subset R[x_1, x_2, \ldots, x_n]$ a finite set $G \subset R[x_1, x_2, \ldots, x_n]$ is called a **Gröbner basis** of I (see Definition 1.3.6)if

$$G \subset I, \text{ and } \mathrm{L}(I) = \mathrm{L}(G).$$

That is, G is a Gröbner basis, if the leading terms of G generate the leading ideal of I. Note that in general, all defined objects depend on the chosen monomial ordering. Especially, a set G may be a Gröbner basis only with respect to a certain monomial ordering. We also remind the reader that with the given definition, G already generates I, as proved in Lemma 1.3.9.

G is furthermore called a **strong Gröbner basis** if for any $f \in I \backslash \{0\}$ there exists a polynomial $g \in G$ satisfying $\mathrm{LT}(g) \,|\, \mathrm{LT}(f)$. A strong Gröbner basis G is called **minimal strong** if $\mathrm{LT}(g_1) \nmid \mathrm{LT}(g_2)$ for all distinct $g_1, g_2 \in G$. A strong Gröbner basis can always be computed when R is a principal ideal ring as shown in Section 2.2.

Note that if R is a field, any non-zero coefficient of a term is invertible in R, and thus $\mathrm{L}(A) = \langle \mathrm{LM}(f) \mid f \in A \rangle$. It is easy to verify that in this case every Gröbner basis is a strong Gröbner basis. As the following example shows, this does in general not hold when R is a ring:

Example 4.2.1. *Consider $R := \mathbb{Z}/6$, and the polynomial ring $R[x]$ with one variable. Then $G := \{2x, 3x\}$ is a Gröbner basis of the ideal $I := \langle x \rangle$. But since neither $2x$ nor $3x$ divide x, G is not a strong Gröbner basis.*

We shall now capture the central notions of this chapter.

Definition 4.2.2. *To any polynomial $f \in R[x_1, x_2, \ldots, x_n]$ we associate the polynomial function $\widetilde{f} : R^n \to R$, $(c_1, c_2, \ldots, c_n) \mapsto f(c_1, c_2, \ldots, c_n)$. We call f a **ideal of vanishing polynomial** if the function $\widetilde{f}$ is identically zero.*

*The set $N_R[\mathbf{x}] = \{f \in R[x_1, x_2, \ldots, x_n] \mid f \text{ is a vanishing polynomial}\}$ is obviously an ideal in $R[x_1, x_2, \ldots, x_n]$, called the **ideal of vanishing polynomials**.*

4.3. A Minimal Strong Gröbner Basis of the Ideal of Vanishing Polynomials

4.3.1. The Ideal of Vanishing Polynomials

From now on let the coefficient ring be $R = \mathbb{Z}/m$, where $m \geq 2$, except stated otherwise. The following results were inspired by the work of Singmaster [74], Kempner [44], Halbeisen and Hungerbühler [38], and Hungerbühler and Specker [43]. Already in Lemma 5 of [44], a univariate version of the following lemma was proven. Theorem 7 of [38] restated this result, and [43] came up with a generalization to multivariate polynomial rings over $\mathbb{Z}/m$.

Lemma 4.3.1. *Let $a \in \mathbb{Z}$ and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$ such that $m|_{\mathbb{Z}} a\alpha!$. Then*

$$p_{\alpha,a} := a \prod_{i=1}^{n} \prod_{l=1}^{\alpha_i} (x_i - l) \in \mathbb{Z}/m[x_1, \ldots, x_n]$$

is a vanishing polynomial.

Proof. Fix an arbitrary point $(c_1, c_2, \ldots, c_n) \in C^n$. Then $p_{\alpha,a}(c_1, c_2, \ldots, c_n)$ contains, for all i, by definition the α_i successive factors $c_i - 1, c_i - 2, \ldots, c_i - \alpha_i$. Independent of the value of c_i, these contain all factors from 2 up to α_i. Therefore, $\alpha_i!$ divides $p_{\alpha,a}(c_1, c_2, \ldots, c_n)$, for all i. By combining these results, it follows immediately that $a\alpha_1! \cdots \alpha_n!$ divides $p_{\alpha,a}(c_1, c_2, \ldots, c_n)$. With $m|_{\mathbb{Z}} a\alpha!$ this yields $p_{\alpha,a}(c_1, c_2, \ldots, c_n) = 0$ modulo m. □

Let us now take a closer look at an arbitrary vanishing polynomial:

Lemma 4.3.2. *Let $f \in I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ be an arbitrary vanishing polynomial with* $\mathrm{LT}(f) = b\mathbf{x}^\beta$. *Then* $m|_{\mathbb{Z}}\, b\beta!$.

For the proof we use some of the ideas introduced in [43], which are based on the notion of partial differences in the multivariate setting. Already Carlitz used partial differences in the univariate case, see [17], to give a necessary and sufficient condition for a function f over $\mathbb{Z}/p^k$ to be a polynomial function (i. e., $f(a) = g(a) \bmod p^k$, for all $a \in \mathbb{Z}/p^k$ and some polynomial $g \in \mathbb{Z}/p^k[x]$).

Proof. Let $R[x_1, \ldots, x_n]$ denote an arbitrary polynomial ring over $n \geq 1$ variables, and let $h \in R[\mathbf{x}]$ be a polynomial. Then we may define the i^{th} partial difference

$$\nabla_i h := h(x_1, \ldots, x_{i-1}, x_i + 1, x_{i+1}, \ldots, x_n) - h(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n),$$

for $1 \leq i \leq n$. Note that ∇_i is a linear operator.

Now we can define the successive application of the operator by

$$\nabla_i^0 h := h, \quad \text{and} \quad \nabla_i^{k+1} h := \nabla_i \nabla_i^k h, \quad \text{for } k \geq 0.$$

(For n = 1, $\nabla_1^k h$ coincides with Carlitz' $\triangle^k h$; see [17].)

Since obviously,

$$\begin{aligned}\nabla_i \nabla_j h = h(x_1, \ldots, x_i + 1, \ldots, x_j + 1, \ldots, x_n) - h(x_1, \ldots, x_i + 1, \ldots, x_n) \\ - h(x_1, \ldots, x_j + 1, \ldots, x_n) + h(x_1, \ldots, x_n) = \nabla_j \nabla_i h,\end{aligned}$$

for all $i, j \in \{1, \ldots, n\}$, we can extend the operator to arbitrary multi-indices, that is, with $\alpha = (\alpha_1, \ldots, \alpha_n) \in \{0, 1, 2, \ldots\}^n$, the term

$$\nabla^\alpha h := \nabla_1^{\alpha_1} \nabla_2^{\alpha_2} \ldots \nabla_n^{\alpha_n} h$$

is independent from the order of application of the ∇_i operators and hence well-defined.

Let us consider the difference $(x_i + 1)^k - x_i^k = k \cdot x_i^{k-1} + g(x_i)$, where g consists of lower terms only, that is, $\deg(g) < k - 1$. A simple induction shows that $\nabla_i^k x_i^k = k!$ and $\nabla_i^j x_i^k = 0$, whenever $j > k$. Let now $a\mathbf{x}^\alpha := \mathrm{LT}(h)$ denote the leading term. Then, mainly due to the linearity of the ∇_i operators, it is easy to see that the previous facts can be further abstracted to the general statements

$$\nabla^\alpha h = a\alpha! \quad \text{and} \quad \nabla^\beta h = 0, \quad \text{for all } \beta \succ \alpha.$$

We apply the first equation to the vanishing polynomial f over the ring $\mathbb{Z}/m$: With f also $\nabla^\beta f = b\beta!$ must be a vanishing polynomial, by construction. But this implies $b\beta! = 0$ modulo m. □

4.3.2. A Minimal Strong Gröbner Basis of I_0

The above lemmas suggest to consider the set of all polynomials $p_{\alpha,a}$ for which neither α nor a can be replaced by a smaller multi-index or element of $\mathbb{Z}/m$, respectively, without loosing the condition $m|_{\mathbb{Z}} a\alpha!$. (This minimality of α has been inspired by the so-called Smarandache function which maps m to $min\{k \in \mathbb{N} \mid m|_{\mathbb{Z}} k!\}$. This function played a role in previous works which studied the univariate case, and had been named after F. Smarandache, see [75], although the idea had been introduced earlier by Kempner in Definition 1 of [44].) We thus define

$$\begin{aligned} S_m := \{\, (\alpha, a) \mid 1 \le a < m,\ a|_{\mathbb{Z}} m,\ \alpha \in \mathbb{N}_0^n,\ m|_{\mathbb{Z}} a\alpha!, \\ \forall\, \beta \prec \alpha\ :\ m \nmid_{\mathbb{Z}} a\beta!, \\ \forall\, b < a,\, b|_{\mathbb{Z}} a\ :\ m \nmid_{\mathbb{Z}} b\alpha! \,\}, \\ G_m := \{\, p_{\alpha,a} \mid (\alpha, a) \in S_m \,\}. \end{aligned}$$

Note that, according to Lemma 4.3.1, all polynomials in G_m will still be elements of I_0. And by Lemma 4.3.2, we can hope to have constructed a strong Gröbner basis.

Theorem 4.3.3. *Let $m \ge 2$ and $n \ge 1$ be arbitrary integers. With the above notations, G_m is a minimal strong Gröbner basis of the ideal of vanishing polynomials $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$, independent of the global monomial ordering.*

Before we prove the theorem, let us take a look at an example.

Example 4.3.4. *Let $m = q_1 \cdot q_2 \cdots q_k$ be a product of $k \ge 1$ mutually distinct primes, and $n \ge 1$ arbitrary. We assume $q_1 < q_2 < \ldots < q_k$. Then we can immediately write down all elements of G_m:*

$$\begin{aligned} (x_i - 1)(x_i - 2) \cdots (x_i - q_k), \\ q_k \cdot (x_i - 1)(x_i - 2) \cdots (x_i - q_{k-1}), \\ q_k \cdot q_{k-1} \cdot (x_i - 1)(x_i - 2) \cdots (x_i - q_{k-2}), \\ \cdots \\ q_k \cdot q_{k-1} \cdots q_2 \cdot (x_i - 1)(x_i - 2) \cdots (x_i - q_1), \end{aligned}$$

in each row for all $i \in \{1, 2, \ldots, n\}$.

Note that the first type of polynomial is in G_m, as $q_k!$ already contains all q_j, thus $m|_{\mathbb{Z}} q_k!$. Also, we need to have all q_k polynomial factors since, for all $r < q_k$, $q_k \nmid_{\mathbb{Z}} r!$, i. e. $m \nmid_{\mathbb{Z}} r!$. For the following polynomials, the argument is similar. Moreover, it is easy to see that we do not have elements in G_m involving two or more variables, and the presented polynomials are all elements of G_m.

In this special case $|G_m| = k \cdot n$, and the maximal degree is q_k. This means that the size of the basis is only linear in the number of variables.

For the case $k = 1$, $\mathbb{Z}/q_1$ is a field, and we obtain only the n polynomials in the top row, which are well-known for this case.

We now prove the theorem:

Proof. Let us fix $m \geq 2$, the number of variables $n \geq 1$, and an arbitrary global monomial ordering. We first show that G_m is indeed a Gröbner basis of I_0. To this end, it suffices to show that (i) S_m and hence G_m is a finite set, (ii) $G_m \subset I_0$, and (iii) $\mathrm{L}(I_0) \subset \mathrm{L}(G_m)$, since (ii) implies the other inclusion $\mathrm{L}(G_m) \subset \mathrm{L}(I_0)$.

(i) Since $(\alpha, a) \in S_m$ implies $\alpha \preceq (m, m, \ldots, m)$, the set is clearly finite.

(ii) G_m consists of polynomials $p_{\alpha,a}$ with $m|_{\mathbb{Z}} a\alpha!$. Then $G_m \subset I_0$ by Lemma 4.3.1.

(iii) Let $f \in \mathrm{L}(I_0)$ be arbitrary. Then there exist some integer $N \geq 1, h_i \in \mathbb{Z}/(m)[x_1, x_2, \ldots, x_n]$ and $f_i \in I_0, 1 \leq i \leq N$, such that

$$f = \sum_{i=1}^{N} h_i \cdot \mathrm{LT}(f_i).$$

Writing $a_i \mathbf{x}^{\alpha^{(i)}} := \mathrm{LT}(f_i)$, we obtain $m|_{\mathbb{Z}} a_i \alpha^{(i)}!$ from Lemma 4.3.2. Now either $(\alpha^{(i)}, a_i)$ is already an element of S_m. Or we can replace a_i by some $b_i|_{\mathbb{Z}} a_i$ and/or $\alpha^{(i)}$ by some $\beta^{(i)} \preceq \alpha^{(i)}$ such that $(\beta^{(i)}, b_i) \in S_m$. We can subsume both cases in saying that, for each $i \in \{1, 2, \ldots, N\}$, there is some $(\beta^{(i)}, b_i) \in S_m$ such that $b_i \mathbf{x}^{\beta^{(i)}} | \mathrm{LT}(f_i)$. With appropriate polynomials $g_i, 1 \leq i \leq N$, this amounts to

$$f = \sum_{i=1}^{N} h_i \cdot g_i \cdot \mathrm{LT}\left(p_{\beta^{(i)}, b_i}\right),$$

i. e., $f \in \mathrm{L}(G_m)$.

Next, let $f \in I_0$. Then, with the same argument as for the f_i above, there exists a $p_{\gamma,c} \in G_m$ such that $\mathrm{LT}(p_{\gamma,c}) | \mathrm{LT}(f)$. This shows that G_m is a strong Gröbner basis.

It remains to show that G_m is minimal. To this end, pick two pairs $(\alpha, a), (\beta, b) \in S_m$ such that $a\mathbf{x}^{\alpha} | b\mathbf{x}^{\beta}$. Then $a|_m b$, $a|_{\mathbb{Z}} m$, $b|_{\mathbb{Z}} m$, and $\alpha \preceq \beta$. We need to prove that $a = b$ and $\alpha = \beta$. Computing in $\mathbb{Z}$, take a prime factor q of b and $k \geq 1$ maximal such that $q^k|_{\mathbb{Z}} b$. Suppose $q^k \nmid_{\mathbb{Z}} a$. Then $a\alpha!$ would have at least one less factor q in its prime factorization than $b\alpha!$. But since $m|_{\mathbb{Z}} a\alpha!$, we then had $m|_{\mathbb{Z}} b/q \cdot \alpha!|_{\mathbb{Z}} b/q \cdot \beta!$, and b would not be minimal in $(\beta, b) \in S_m$. We conclude that $b|_{\mathbb{Z}} a$. We write this as $a = d \cdot b$ for some $d|_{\mathbb{Z}} m$. Now $a|_m b$, that is, $m|_{\mathbb{Z}} a \cdot c - b$ for some c. Putting things together we get $bd = a|_{\mathbb{Z}} m|_{\mathbb{Z}} bcd - b = b(cd - 1)$. Hence $d|_{\mathbb{Z}} (cd - 1)$ which can only hold for $d = 1$, implying $a = b$. But then we must also have $\alpha = \beta$, since otherwise β would not be minimal in $(\beta, b) \in S_m$. □

We show that leading terms of a min. strong Gröbner bases of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ are unique, up to multiplication by units of $\mathbb{Z}/m$. We prove this result as a consequence of a more general statement for ideals over arbitrary commutative rings with 1 that has, to our knowledge, not been stated before. (Note the similar statement in the field case; see e.g. Proposition 1.8.4 in [1].)

Theorem 4.3.5. *a) Let G, F be two minimal strong Gröbner basis of an arbitrary ideal $I \subset R[x_1, x_2, \ldots, x_n]$, where R is any commutative ring with 1. Then $|G| = |F|$, and the sets of leading terms in G and F coincide up to multiplication by elements of R, i. e.,*

$$\forall\, g \in G \;\; \exists\, f \in F \;\; \exists\, c \in R \;\; \mathrm{LT}(g) = c \cdot \mathrm{LT}(f). \quad (*)$$

b) In the case of $R = \mathbb{Z}/m$ and $I = I_0$, the ring elements c in $(*)$ can be chosen to be units of $\mathbb{Z}/m$.

Note that the second statement holds for any ideal, if the ring R is a domain.

Proof. a) Starting with the proof of $(*)$, we pick any $g \in G \subset I$. Then, by strongness of F, there is some $f \in F$ such that $\mathrm{LT}(f) \,|\, \mathrm{LT}(g)$. Vice versa, by strongness of G, there must be some $g' \in G$ such that $\mathrm{LT}(g') \,|\, \mathrm{LT}(f)$. Therefore, $\mathrm{LT}(g') \,|\, \mathrm{LT}(f) \,|\, \mathrm{LT}(g)$, which implies $g = g'$, by minimality of G. But then the leading monomials $\mathrm{LM}(f)$ and $\mathrm{LM}(g)$ must also coincide, yielding the desired relation between $\mathrm{LT}(f)$ and $\mathrm{LT}(g)$.

Similar to the previous argument, it is easy to see that no two distinct leading terms in F can fulfil a relation $(*)$ with the same leading term in G, and vice versa. This implies the equality $|\{\mathrm{LT}(g) \mid g \in G\}| = |\{\mathrm{LT}(f) \mid f \in F\}|$ which clearly amounts to $|G| = |F|$, by the minimality of G and F.

b) We first choose $G = G_m$ to be the explicitly given Gröbner basis, and F any other minimal strong Gröbner basis of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$. Consider a relation as in $(*)$, i. e., $b \cdot \mathbf{x}^\beta = c \cdot a \cdot \mathbf{x}^\alpha$, where $(\beta, b) \in S_m$ and $a \cdot \mathbf{x}^\alpha$ denotes the leading term of some $f \in F$. Then $b = a \cdot c \bmod m$, in other words $m|_{\mathbb{Z}}\, ac - b$. Now let $\tilde{a} := \gcd(a, m)$ be the maximum portion of a that divides m, that is, $a = \tilde{a} \cdot u$, where $\gcd(u, m) = 1$ which is equivalent to u being a unit in $\mathbb{Z}/m$. Since $\tilde{a}|_{\mathbb{Z}}\, m|_{\mathbb{Z}}\, ac - b$, we obtain $\tilde{a}|_{\mathbb{Z}}\, b$.

We want to show $\tilde{a} = b$, so for a contradiction let us assume $\tilde{a} < b$. $f \in F \subset I_0$ implies $m|_{\mathbb{Z}}\, a\alpha!$ by Lemma 4.3.2, hence $m|_{\mathbb{Z}}\, \tilde{a}\alpha! = \tilde{a}\beta!$, as the factors in $a/\tilde{a}$ do not affect divisibility by m and since obviously $\alpha = \beta$. But this means that we could replace b by the smaller $\tilde{a}$ and still preserve the condition $m|_{\mathbb{Z}}\, \tilde{a}\beta!$. This contradicts the minimality of b in $(\beta, b) \in S_m$. Hence $\tilde{a} = b$.

We thus arrive at the claimed relation $u \cdot b\mathbf{x}^\beta = a\mathbf{x}^\alpha$, and c can be replaced by the unit $u^{-1} \in (\mathbb{Z}/m)^*$.

We have shown that we can relate the leading terms of any minimal strong Gröbner basis F of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ to the leading terms in G_m by units. By transitivity, we can now clearly also relate the leading terms of any two minimal strong Gröbner bases by units. This concludes the proof. □

Note that an arbitrary factor c, relating two leading terms, need not necessarily be a unit. For example, consider the polynomial $f(x, y) = 3(x-1)(x-2)\cdot(y-1)(y-2) \in G_{12}$. We may switch to another minimal strong Gröbner basis of $I_0 \subset \mathbb{Z}/12[x, y]$, simply by replacing $f(x, y)$ by $f'(x, y) = 9(x-1)(x-2) \cdot (y-1)(y-2)$. Note that over $\mathbb{Z}/12$ the ideals $\langle f \rangle$ and $\langle f' \rangle$ are identical. Thus, $G_m \setminus \{f\} \cup \{f'\}$ must still be a minimal strong Gröbner basis. Now obviously $\mathrm{LT}(f') = 3 \cdot \mathrm{LT}(f)$, but 3 is not a unit in $\mathbb{Z}/12$.

We point out that minimal strong Gröbner bases are in general not unique. This is due to the fact that we only consider leading terms and do not require tail reduction here. For example, in the case of the ideal I_0, we can easily modify the basis G_m and still obtain a minimal strong Gröbner basis. To this end, we may pick two elements $f, g \in G_m$ with $\mathrm{LM}(g) < \mathrm{LM}(f)$ and replace f by $f + g$.

Let us once again take a look at the complexity of G_m, that is, the size $|G_m|$ as a function of the number of variables n. The discussion that followed Example 3.4 already

made clear that $|G_m|$ is only linear in n, when all prime factors of m are mutually distinct. In the general case when $m = q_1^{e_1} \cdot q_2^{e_2} \cdots q_k^{e_k}$ with some $e_j > 1$, the construction is combinatorially more complex. However, based on the following investigation for the practically relevant case $m = q^k$, we conjecture that for fixed m the size of G_m is always polynomial in n.

Since we are interested in the asymptotic behavior of $|G_m|$ for large n, we may assume that n is much larger than $m = q^k$. We can decompose G_m into the disjoint union

$$\begin{aligned} G_m &= \bigcup_{0 \le j < k} G_m^{(j)}, \text{ where} \\ G_m^{(j)} :=& \{q^j \cdot (x_i - 1) \cdots (x_i - (k-j)q) \mid 1 \le i \le n\} \\ &\cup \{q^j \cdot (x_{i_1} - 1) \cdots (x_{i_1} - s_1 q)(x_{i_2} - 1) \cdots (x_{i_2} - s_2 q) \mid \\ &\quad 1 \le i_1, i_2 \le n; i_1 \ne i_2; 1 \le s_1, s_2; s_1 + s_2 = k - j\} \\ &\cdots \\ &\cup \{q^j \cdot (x_{i_1} - 1) \cdots (x_{i_1} - q)(x_{i_2} - 1) \cdots (x_{i_2} - q) \cdots \\ &\quad (x_{i_{k-j}} - 1) \cdots (x_{i_{k-j}} - q) \mid 1 \le i_u \le n; i_u \ne i_v \text{ for } u \ne v\}, \end{aligned}$$

that is, in $G_m^{(j)}$ we have the constant coefficient q^j, and we have polynomials in 1 up to $k - j$ variables. With $h_j := |G_m^{(j)}|$, we obtain the very rough estimates

$$h_j \le n + \binom{n}{2} \cdot k^1 + \cdots + \binom{n}{k-j} \cdot k^{k-j-1} = \sum_{l=1}^{k-j} \binom{n}{l} \cdot k^{l-1} \le \binom{n}{k} \cdot k^k,$$

$$h_j \ge \binom{n}{k-j}.$$

For $h := |G_m| = \sum_{0 \le j < k} h_j$ we thus get

$$\binom{n}{k} \le \sum_{j=0}^{k-1} \binom{n}{k-j} \le h \le k \cdot \binom{n}{k} \cdot k^k = \binom{n}{k} \cdot k^{k+1},$$

and $h = |G_m|$ is polynomial of degree k in the number of variables n.

4.3.3. Computing the Reduced Normal Form of a Polynomial

After we have given a minimal strong Gröbner basis of $I_0 \subset \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$, we shall now turn to computing representatives of the residue classes in $(\mathbb{Z}/m[x_1, x_2, \ldots, x_n])/I_0$. When we impose certain bounds on the coefficients of all monomials, these representatives are unique:

Proposition 4.3.6. *Every residue class $\bar{f} \in (\mathbb{Z}/m[x_1, x_2, \ldots, x_n])/I_0$ has a unique representative $f \in \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ of the form*

$$f = \sum_{\alpha \in \{0,1,\ldots,m-1\}^n} a_\alpha \mathbf{x}^\alpha, \text{ where } 0 \le a_\alpha < \frac{m}{\gcd(m, \alpha!)}, \text{ for all } \alpha.$$

Note that, whenever $m|_{\mathbb{Z}}\, \alpha!$, the given bound forces a_α to be zero.

Proof. Let $f \in \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ be an arbitrary polynomial. Suppose f contains a monomial $a\mathbf{x}^\alpha$ for which $a \geq c := \frac{m}{\gcd(m,\alpha!)}$. Due to division with remainder of a by c in $\mathbb{Z}$, we obtain $a = k \cdot c + r$ for some $k \in \{1, 2, \ldots\}$, and $0 \leq r < c$. Now, $m|_{\mathbb{Z}}\, \frac{m\alpha!}{\gcd(m,\alpha!)}$. In other words, $m|_{\mathbb{Z}}\, c\alpha!$, and $p_{\alpha,c} \in I_0$ by Lemma 4.3.1.

As a consequence, f and $f' := f - k \cdot p_{\alpha,c}$ lie in the same residue class. Moreover, the coefficient of $\mathbf{x}^\alpha$ in f' is $a - k \cdot c = r$, for which the claimed bound holds. Since we have a global ordering on the monomials, we need only finitely many repetitions of the presented reduction step, in order to arrive at a polynomial g which also lies in the residue class of f, and the coefficients of which all satisfy the required bound condition.

For proving uniqueness of the constructed representative, assume we have two representatives f_1, f_2 of the residue class of f, realizing all coefficient bounds. Then, by defining either $g := f_1 - f_2$ or $g := f_2 - f_1$, we obtain a polynomial $g \in I_0$ with $\mathrm{LT}(g) = a\mathbf{x}^\alpha$ and $0 \leq a < \frac{m}{\gcd(m,\alpha!)}$. By Lemma 4.3.2, we know that $m|_{\mathbb{Z}}\, a\alpha!$. We need to show that $a = 0$; so for a contradiction, let us assume that $a > 0$. With $b := \gcd(m, a)$ we still have $m|_{\mathbb{Z}}\, b\alpha!$, i. e., $\frac{m}{b}|_{\mathbb{Z}}\, \alpha!$. Then also $\frac{m}{b}|_{\mathbb{Z}}\, \gcd(m, \alpha!)$ which implies $m|_{\mathbb{Z}}\, b \cdot \gcd(m, \alpha!)$. But $b \cdot \gcd(m, \alpha!) \leq a \cdot \gcd(m, \alpha!) < m$, yielding the desired contradiction. □

As an immediate consequence, we can count the number of polynomial functions which is the same as the number of residue classes in $(\mathbb{Z}/m[x_1, x_2, \ldots, x_n])/I_0$:

Corollary 4.3.7. *The number of polynomial functions $(\mathbb{Z}/m)^n \to \mathbb{Z}/m$ is given by*

$$N = \prod_{\alpha \in \{0,1,\ldots,m-1\}^n} \frac{m}{\gcd(m, \alpha!)}.$$

In comparison, the number of all functions $(\mathbb{Z}/m)^n \to \mathbb{Z}/m$ equals

$$m^{(m^n)} = \prod_{\alpha \in \{0,1,\ldots,m-1\}^n} m = N \cdot \prod_{\alpha \in \{0,1,\ldots,m-1\}^n} \gcd(m, \alpha!)\,.$$

$\mathbb{Z}/m \longrightarrow \mathbb{Z}/m$	No. of functions	No. of polynomial functions
$m = 2^2$	256	64
$m = 2^8$	10^{616}	10^{16}
$m = 2^{16}$	10^{315652}	10^{52}
$m = 2^{32}$	$10^{41373247567}$	10^{184}

Hence, if m is not prime, there are fewer polynomial functions $(\mathbb{Z}/m)^n \to \mathbb{Z}/m$ than functions. This has the consequence that not every problem which can be modeled by functions, like problems coming from formal verification, can be modeled by polynomials over $\mathbb{Z}/m$ (cf. [89] where, nevertheless, polynomial ideals over $\mathbb{Z}/2^k$ have been used successfully).

The following conjecture was verified for small m, n.

Conjecture 4.3.8. *A function $\mathbb{Z}_m^n \to \mathbb{Z}_m$ is polynomial if and only if Newton interpolation works. This means that the division during the algorithm is possible, but not necessarily unique.*

Further a straightforward generalization of the Nullstellensatz is not possible as the following lemma shows.

Lemma 4.3.9. *Let R be a ring with zero divisors. There exists no ring $\hat{R} \supset R$, such that every non-constant polynomial of $R[x]$ has a zero in $\hat{R}$.*

Proof. Let $n \in R\backslash\{0\}$ be a zero divisor and consider $f = nx - 1$. Assume there exists a ring $\hat{R} \supset R$ which contains a root r of f. Then $f(r) = n \cdot r - 1 = 0$ and hence $1 = n \cdot r$. On the other hand, there exists an $m \neq 0$ with $m \cdot n = 0$ and hence $m \cdot 1 = m \cdot n \cdot r = 0$, a contradiction. □

Following the idea in the proof of Proposition 3.6, we are able to present a very fast algorithm for computing the reduced normal form, that is, the unique representative of a residue class in the ring $\mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ module I_0. (see [72] for $\mathbb{Z}/2^k$):

Algorithm 8 Reduced normal form in $\mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ with respect to I_0

Input: $f \in \mathbb{Z}/m[x_1, x_2, \ldots, x_n]$ a polynomial
Input: $>$ any monomial ordering on $\mathbb{Z}/m[x_1, x_2, \ldots, x_n]$
Output: h the reduced normal form of f with respect to I_0

```
h := 0
while f ≠ 0 do
  a x^α := LT(f)
  c := m / gcd(m, α!)
  solve a = k · c + r with k ∈ ℕ and 0 ≤ r < c
  h := h + r x^α
  f := f − k · p_{α,c} − r x^α
end while
return h
```

Note that the algorithm makes sure that $f + h$ will always represent the same residue class, as $p_{\alpha,c} \in I_0$. Since initially $h = 0$, this class must be the residue class of f. After termination, which is ensured by the global ordering, h consists only of terms with appropriate coefficient bound, i. e., h must be the unique representative as given in Proposition 3.6.

4.3.4. Computing Minimal Strong Gröbner Bases over Different Rings $\mathbb{Z}/m$

The simple structure of minimal strong Gröbner bases provides us with a recursive means to construct G_m from bases for smaller m. We are especially interested in computing G_M from the elements of the already computed set G_m, where $M = q \cdot m$ with q a prime number. The following pairwise disjoint decomposition of G_M is easy to verify:

$$\begin{aligned} G_M = &\{p_{\alpha,a} \mid p_{\alpha,a} \in G_m, (\alpha, a) \in S_M\} \\ &\cup \{p_{\alpha,aq} \mid p_{\alpha,a} \in G_m, (\alpha, aq) \in S_M\} \\ &\cup \{p_{\alpha+\beta,b} \mid p_{\alpha,a} \in G_m,\ \exists\, \beta \in B(\alpha, a, q)\ \exists\, b|_{\mathbb{Z}} M\ :\ (\alpha+\beta, b) \in S_M\}, \end{aligned}$$

where $B(\alpha, a, q)$ denotes the set of all $\beta \succ (0, 0, \ldots, 0)$ such that $(\alpha + \beta)!$ contains one more prime factor q than $a\alpha!$.

Algorithm 9 RecComp(M), Recursive computation of G_M

Input: $M \in \{2, 3, \ldots\}$
Output: G_M

```
pick any prime factor q of M
if M = q then
  A := {q · e_i | 1 ≤ i ≤ n}, where the e_i are the unit vectors in ℕ^n
  G := {p_{α,1} | α ∈ A}
else
  m := M/q
  H :=RecComp(m)
  G := { }
  for all p_{α,a} ∈ H do
    if (α, a) ∈ S_M then
      G := G ∪ {p_{α,a}}
    else
      G := G ∪ {p_{α,a·q}}
      for all β ∈ B(α, a, q) ⊂ {β | (0, 0, ..., 0) ≺ β ⪯ (q, q, ..., q)} do
        b := M / gcd(M, (α+β)!)
        G := G ∪ {p_{α+β,b}}
      end for
    end if
  end for
end if
return G
```

This decomposition says that we may already directly find elements of G_M in G_m. Or, secondly, we may build an element of G_M by multiplying an element of G_m by q. Besides altering the coefficient only, we can also try to enlarge the exponent vector of some $p_{\alpha,a} \in G_m$ such that the new exponent factorial $(\alpha + \beta)!$ contains one more prime

factor q than $a\alpha!$. However, enlarging the exponent may introduce many more divisors of M, so that in general we need to adjust the coefficient. It is easy to see that once a suitable β is found, we can set $b = \frac{M}{\gcd(M,(\alpha+\beta)!)}$. The search for suitable β can obviously be limited to the set defined by the condition $\beta \preceq (q, q, \ldots, q)$, that is, we know a finite superset of $B(\alpha, a, q)$.

In practice, all three cases may occur. The following examples are numbered according to the order in the above decomposition. (The number of variables, n, equals 2.)

Example 4.3.10.

1. *$G_3 \subset G_6$, since $3! = 6$ already contains all necessary factors; see Example 3.4 (and the remark regarding $k = 1$) to recall the elements of G_3.*

2. *With q any prime, we have $p_{(3,0),2} \in G_{12}$ and $p_{(3,0),2\cdot q} \in G_{12\cdot q}$.*

3. *We have $6(x-1)(x-2)(y-1)(y-2) \in G_{24}$. We try to construct an element in $G_{24\cdot 3}$ by enlarging the product of x and y terms. Since $6 \cdot 2! \cdot 2!$ contains one prime factor 3, we try to move to the target product $(x-1)(x-2)(x-3)(y-1)(y-2)(y-3)$ which realizes one more factor 3 because $3^2|_{\mathbb{Z}}\, 3! \cdot 3!$. Now $b = \frac{72}{\gcd(72,3!\cdot 3!)} = 2$ and hence $2(x-1)(x-2)\cdot(x-3)(y-1)(y-2)(y-3) \in G_{72}$.*

The above decomposition of G_M, and the structure of G_q for a prime q as discussed in Example 3.4, give rise to the following algorithm.

Chapter 5.

Applications to formal verification

A new approach for proving arithmetic correctness of data paths in System-on-Chip modules is described in Section 5.3 after preliminaries and further investigated approaches in the two prior sections. Our new approach complements existing techniques which are, for reasons of complexity, restricted to verifying only the control behavior. Normalization at the arithmetic bit level (ABL) is combined with the techniques of computer algebra to verify arithmetic parts of modern microchip designs. The approach proves tractable for industrial data path designs where standard property checking techniques fail (e. g. Infineon's Tricore 2).

The joint research was done and continues to grow within the project "Verification of Systems-on-Chip with algebraic Methods (VerSys)" of the Center for Mathematical and Computational Modeling $(CM)^2$ [19].

This chapter is based on several publications [89, 9, 62] together with Michael Brickenstein, Alexander Dreyer, Gert-Martin Greuel, Wolfgang Kunz, Evenly Paleness, Dominie Scoffed and Markus Wedler.

5.1. Introduction

Property checking has become well-established in modern design flows for Systems-on-Chip (SoCs). Its main application domain is ensuring the correctness of the individual SoC blocks. This does not only lead to high quality IP (intellectual property) modules but also reduces the costs for system integration and chip level simulation. Given IP modules of provably high quality, chip level simulation may concentrate on true system level aspects and is relieved from hunting bugs in local modules. Therefore, in recent years, a lot of effort has been made to develop sophisticated methodologies and tools for formal module verification based on property checking. Today, formal property checking can handle almost all types of modules that can be found in today's SoCs. Nonetheless, a few pathological cases remain that sometimes limit the application of property checking in industrial practice. In particular, data paths are often a challenge for formal techniques, especially, if not only the correctness of the control flow but also correctness of the data is to be proved.

For complex arithmetic data paths simulation is, therefore, still prevailing in industrial verification environments. This is due to the inability of standard proving procedures based on satisfiability solving (SAT) or binary decision diagrams (BDDs) to handle

arithmetic functions. Especially multiplication — as it is part of nearly all data paths for signal processing applications — has remained a severe problem for standard tools. This deficiency has motivated the research community to investigate alternative proof methods with focus on arithmetic.

In case the validity of a property can be proven without consideration of the exact functionality of the data path, abstraction and refinement techniques have shown superiority over pure Boolean SAT techniques. A survey on these techniques can be found in [45]. However, for properties that depend on the exact functionality of the datapath a suitable abstraction is not likely to be found.

Another direction of research investigates SAT-modulo-theory (SMT) solvers. These solvers combine a SAT solver with specialized solvers for certain well-selected theories. An example for such a theory is the theory of equality with uninterpreted functions used in UCLID [69]. In case the problem at hand really depends on the exact functionality of a datapath, as is typically the case, most SMT solvers resort to bit blasting [45] for the corresponding problem parts. In this case SMT solvers show the same performance limitations as pure SAT solvers as soon as these datapaths include multiplication operations. The decision problems in RTL-property checking could be expressed as SAT problems for formulas of the quantifier free logic (QF-BV) and in principle be solved using solvers such as Yices [28], MathSat[15], Z3 [22] or Spear [4]. For sophisticated datapath implementations involving multiplication, however, experience shows that the problems are still beyond the capacity of such solvers.

Recently, techniques from symbolic computer algebra have entered the verification arena. The authors of [73] present a procedure to determine whether a multivariate polynomial with fixed word length operands is vanishing. By this means a comparison of polynomial representations for bit vector functions is feasible. This procedure is extended towards multiple word length operands in [70, 71]. However, both approaches require a word level representation of the datapaths under comparison. This limits their applicability in RTL property checking. Due to performance and area requirements RTL designers typically design specialized arithmetic components. These components are often designed using bit level arithmetic circuitry to build addition trees and partial products. The smallest entities in an addition tree can be described using half and full adders in general. An approach for verification of such bit level implementations using Gröbner basis theory over fields is reported in [86]. This approach requires polynomial specifications for every building block in the hierarchy of the arithmetic circuit design. After proving that a block, e.g., a CSA adder, fulfills its local specification, the polynomial representation is used to verify the block in the next level of the hierarchy. However, as the correctness proof includes a range check the intermediate results at the block boundary are required to have sufficient bit width to represent every possible result. For designs implementing integer arithmetic with fixed bit width this is often not the case.

A heuristic approach to exploit the availability of arithmetic bit level (ABL) information in RTL designs has been reported in [87]. In this work a data structure called ABL description for representation of addition networks and bitwise multiplication is transformed into a reduced normal form. By canceling out common addends from addi-

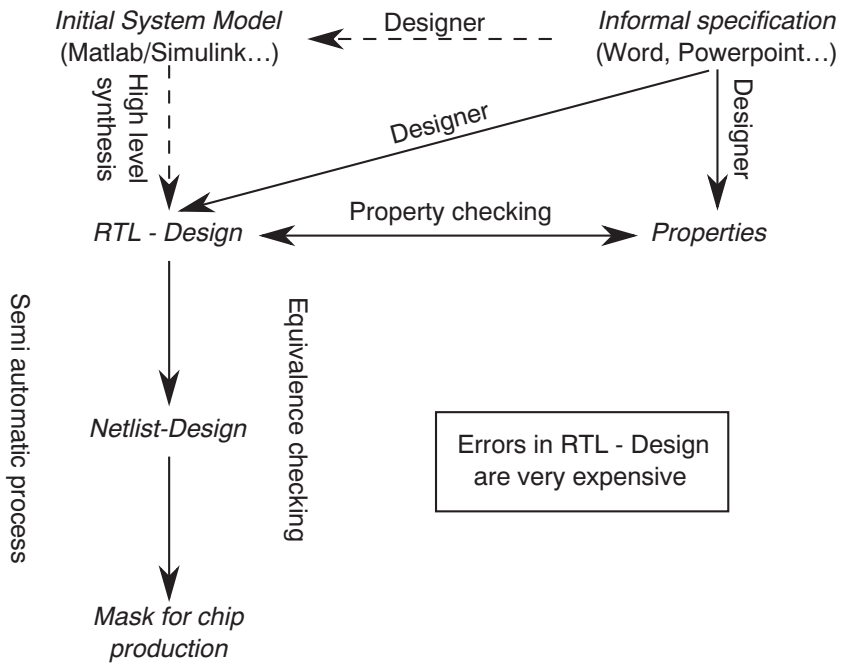


Figure 5.1.: Digital system design flow

tion networks in the fanin of a comparator the normalization approach relieves the SAT solver from reasoning in structurally different implementations for the same arithmetic function.

In order to overcome the limitations of [86] we use computer algebra algorithms for rings $\mathbb{Z}/2^N$ to solve decision problems at the arithmetic bit level. This extends the normalization approach of [87] with a clean and well-understood mathematical foundation. We show that an ABL description [87] can directly be transformed into a set of equivalent *variety subset problems*. We exploit the observation that under certain monomial orderings the set G of polynomials generated from the ABL components forms a Gröbner basis of the ideal $I = \langle G \rangle$ generated by these polynomials with special properties. This allows to solve the variety subset problem and hence decide problems at the arithmetic bit level.

5.1.1. Design flow

The circuit design starts with an informal specification of a microchip (Figure 5.1) by some tender documents which are usually given in a human readable text or presentation format. In a first step the specification may be translated in a high-level modeling language. One possibility is to use high level synthesis for generating a **register transfer level** (RTL) design which describes the flow of signals between registers in terms of a hardware description language [76]. But this is rarely used in practise as it does constrain the freedom of the design. Instead, designers manually create the RTL design

in a hardware description language. Concurrently, intended behavior specified by the informal specification is formalized by formal properties. Automatic tools are used to ensure that the RTL design fulfills these conditions.

After passing property checking a netlist is generated semi-automatically from the RTL. The latter is used to derive the actual layout of the chip mask. The validation that different circuit descriptions arising from the last two steps emit the same behavior, is called **equivalence checking**. Since this can be handled accurately, setting of the RTL design is the most crucial part. Errors at this level may become very expensive, as they may lead to unusable chip masks or even defective prototypes. In the following we present to approaches to overcome present verification problems by computer algebra methods using standard bases over rings.

The ability of checking the validity of a proposed design restricts the design itself: a newly introduced design approach may not be used for an implementation as long as its verification cannot be ensured. In particular, this applies to digital systems consisting of combined logic and arithmetic blocks, which may not be treated with specialized approaches. Here, dedicated methods from computer algebra may lead to more generic procedures, which help to fill the design gap.

5.2. Encoding using native ring variables

5.2.1. Problem formulation and encoding in algebra

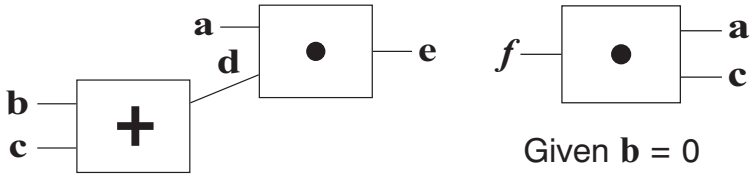
The verification problem is defined by a set of axioms M representing the circuit w. r. t. given decision variables. In addition, a set of statements P represents the property to be checked. For instance, if M models a multiplication unit, a suitable P would be the condition that after a complete cycle the output of M is the product of its inputs.

The question, whether the circuit represented by M fulfills P can be reformulated in the following way: First of all, we may assume, that M is consistent, i. e. there are no contradictions inherent in the axioms, since the axioms describe a circuit. Then the new set of axioms $M \wedge \neg P$ is contradictable if and only if M implies P. Hence the desired property P will be proven by showing, that $M \wedge \neg P$ has no valid instance, i. e. one fulfilling the axioms and not the property.

In the following we encode this logical system into a system of algebraic equations in two ways, on word level and on bit level. The word level model will lead to consider Gröbner bases over the ring $\mathbb{Z}_{2^n}$ while the bit level will lead to Gröbner basis over Boolean rings. Here and in the following $\mathbb{Z}_m$ denotes the finite ring $\mathbb{Z}/m\mathbb{Z}$ for $m \in \mathbb{Z}\backslash\{0\}$.

Word level encoding

We illustrate, how the problem of formal verification can be encoded in a system of algebraic equations using polynomials over the ring $\mathbb{Z}_{2^n}$. Let n be the word length of the circuit, i. e. the number of bits used by each signal (in typical applications we have $n \in \{16, 32, 64\}$). Then the RTL description displayed in Figure 5.2(a) is equivalent



(a) RTL diagram (b) Property

Figure 5.2.: RTL design and property

to the following set of algebraic equations

$$M = \{b + c = d,\ a \cdot d = e\} \tag{5.1}$$

where $b+c-d, a \cdot d - e$ are polynomials in $\mathbb{Z}_{2^n}[a, b, c, d, e, f]$. Of course, the two equations in M are equivalent to $a \cdot (b + c) = e$, but in general the latter input-output form is infeasible due to its complexity. Also, there can be more than one output per block and only some of these outputs may be used further.

For example, Figure 5.2(b) presents the property

$$P = \{b = 0,\ a \cdot c = f\}. \tag{5.2}$$

In this case, the statement that M implies P is equivalent to the assertion that $M \cup P \cup \{f \neq e\}$ has no solution. Since the set $\{f \neq e\}$ is not a closed algebraic set, we replace $f \neq e$ by $s \cdot (f - e) = 2^{n-1}$, where s is a new variable. Indeed, it is easy to see that a value $s \in \mathbb{Z}_{2^n}$ fulfills this equation if and only if $f \neq e$ (since the ring $\mathbb{Z}_{2^n}$ has zero divisors, $f \neq e$ cannot be encoded by $s(f - e) = 1$). Let I be the ideal $\langle \{b + c - d, a \cdot d - e,\ \ b, a \cdot c - f,\ \ s \cdot (f - e) - 2^{n-1}\} \rangle$ in $\mathbb{Z}_{2^n}[a, b, c, d, e, f, s]$. Then the question reduces to the question whether

$$\mathrm{V}(I) := \left\{(a, b, c, d, e, f, s) \in \mathbb{Z}_{2^n}^7 \mid p(a, b, c, d, e, f, s) = 0, \text{ for all } p \in I\right\}$$

is empty. There are no solutions for the ideal I (i. e. $\mathrm{V}(I) = \emptyset$) if and only if $M \wedge \neg P$ is contradictable, that is, P is satisfied by M.

One way of tackling this problem is to compute a Gröbner basis of I in the ring R/I_0, where I_0 denotes the ideal of vanishing polynomials in R, i. e. polynomials evaluating to zero at any point of $\mathbb{Z}_{2^n}^7$. Due to the zero divisors in this ring the ideal I_0 has more structure than in the finite field case and even its Gröbner basis can become huge (cf. Chapter 4).

Bit level encoding

An alternative approach is to encode the problem at the bit level, that is, as polynomials over $\mathbb{Z}_2$. This approach is based on the fact that every value of x in $\mathbb{Z}_{2^n}$ can be encoded uniquely to the base 2, i. e. in its bits:

$$x = x_0 + x_1 2 + \cdots + x_{n-1} 2^{n-1},\ x_i \in \{0, 1\}\,. \tag{5.3}$$

In the example above we can express each variable a, b, c, d, e, f analogously to Equation 5.3 with new variables $a_i, b_i, c_i, d_i, e_i, f_i \in \{0, 1\}, i = 0, \ldots, n-1$. Then Equation 5.1 and Equation 5.2 must be rewritten, which yields n equations for each of them. Gathering all corresponding polynomials and adding the polynomial $\prod(1 - f_i + e_i)$, which is equivalent to $f \neq e$, we obtain an ideal I over $R := \mathbb{Z}_2[a_0, \ldots, f_{n-1}]$ in $6\,n$ variables.

For instance, the bits $p_0, \ldots, p_{n-1} \in \{0, 1\}$ of the product $p = a \cdot b$ are given by equations $p_j = a_j \cdot b_0 + \sum_{i=0}^{j-1}(a_i \cdot b_{j-i} + t_{i,j-i})$ over $\mathbb{Z}_2$, where the $t_{k,l}$ mark rather complicated bit level expressions in the $s_{k,l} \in \{0, 1\}$, which fulfill $p_k + s_{k,1}2 + \cdots + s_{k,n-1}2^{n-1} = a_k \cdot b_0 + \sum_{i=0}^{k-1}(a_i \cdot b_{k-i} + s_{i,k-i})$ in $\mathbb{Z}_{2^n}$. For example, for $n = 4$, we get

$$\begin{aligned}
p_3 &= a_3\,b_0 + a_2\,b_1 + a_1\,b_2 + a_0\,b_3 + a_2\,a_1\,a_0\,b_1\,b_0 + \\
&\qquad a_2\,a_1\,b_1\,b_0 + a_2\,a_0\,b_2\,b_0 + a_1\,a_0\,b_2\,b_1\,b_0 + a_1\,a_0\,b_2\,b_1 + a_1\,a_0\,b_1\,b_0 \\
p_2 &= a_2\,b_0 + a_1\,b_1 + a_0\,b_2 \; + a_1\,a_0\,b_1\,b_0 \\
p_1 &= a_1\,b_0 + a_0\,b_1 \\
p_0 &= a_0\,b_0
\end{aligned}$$

Again let I_0 be the ideal of vanishing polynomials in R. In this case, the ideal I_0 is generated by the field equations $x^2 - x = 0$ for every variable x. Now we compute a Gröbner basis of I in the ring R/I_0. In this ring every ideal is principal and hence its reduced Gröbner basis will consist of just one polynomial. Moreover, $I = \langle 1 \rangle$ if and only if its reduced Gröbner basis is $\{1\}$ and this is equivalent to the zero set of all polynomials in I being empty, and therefore if and only if the property P holds.

Modeling advantages and disadvantages

Both modeling approaches presented in Section 5.2.1 and Section 5.2.1 have strengths and weaknesses. On the one hand, the word level formulation of verification problems as polynomial systems over $\mathbb{Z}_{2^n}$ leads to fewer variables and equations. The equations of arithmetic blocks, like multiplier and adder blocks, are given in a natural and human readable way. However, not all formulæ on word level (for example bitwise `and`, `or`, and `exclusive-or`) may be coded by polynomial equations. Therefore, full strength will need bit level encoding of some variables. Another drawback are the coefficients from $\mathbb{Z}_{2^n}$, which is a ring with zero divisors and not a field. Hence, one cannot rely on valuable properties of fields, like the algebraic closure.

Since $\mathbb{Z}_2$ is a field, these restrictions do not exists for polynomials over $\mathbb{Z}_2$, which can be used for formulation of arbitrary bit level equations. Moreover, since the coefficients are restricted to be one or zero, they need not to be stored at all. Hence, a specialized data structure is possible, which is tailored to suit this application task. On the other hand, contrary to the word level case, bit level formulations carry many variables and equations. The number of them may grow exponentially even for some applications which can be handled easily over $\mathbb{Z}_{2^n}$.

5.2.2. Conclusion

Research and experiments using native ring variables on the word level have shown that the approach is infeasible at the current stage. For details regarding research for bit level encoding please refer to [8].

Instead a further encoding combining certain aspects of the encoding described above have lead to the ability to proof industrial designs. This encoding is described in the next section.

5.3. Encoding using restricted variables

In this section we will describe an encoding inspired by the ABL description (see [87] or the following quick introduction). The encoding with polynomial description over a ring resulted in algorithm feasible to proof correctness of industrial designs. At first a short introduction to ABL descriptions of circuits.

5.3.1. ABL description

Arithmetic bit level (ABL) descriptions as introduced in [87] have proven to be useful for modeling the arithmetic parts of a property checking instance. In this section we briefly review this notion as far as it is required for this document. We use the following notations:

- For $a \in \mathbb{Z}$, $b > 0$ the remainder, $a \mod b$, of the integer division a/b denotes the smallest $k \geq 0$ with $k = a - mb$ for some $m \in \mathbb{Z}$.
- For $n > 0$ and $a \in \mathbb{Z}$ the uniquely determined bit vector $(a_{n-1}, \ldots, a_0)$ with $a \mod 2^n = \sum_{i=0}^{n-1} 2^i a_i$ is denoted as $\langle a, n \rangle = (a_{n-1}, \ldots, a_0)$, i.e., $\langle a, n \rangle$ is the n-bit binary unsigned integer representation of a.
- $\mathbb{B} = \{0, 1\} \subset \mathbb{Z}$ denotes the Boolean space.

The combinatorial transition function of an RTL circuit design is usually modeled by a directed acyclic graph where the vertices are labeled with bit vector functions. It is common practice to translate verification problems for RTL circuits into such bit vector netlists with a single output indicating whether, e.g., a certain property holds for a design. For the arithmetic problem parts we extract an *ABL description* from this netlist. This description again is a directed acyclic graph where the vertices can be of type "partial product generator", "addition network" or "comparator". These vertex types are defined as follows:

Definition 5.3.1. *Let $n, m \in \mathbb{N}$, $w : \{0, \ldots, m\} \to \mathbb{Z}$ and $c \in \mathbb{Z}$. The bit vector function $r : \mathbb{B}^m \to \mathbb{B}^n$ with*

$$r(x_1, \ldots, x_m) = \langle (c + \sum_{i=0}^{m-1} w(i) \cdot x_i), n \rangle$$

is called ***addition network with addend set*** $A = \{x_1, \ldots, x_m\}$. *n is called* result width*, c is called* constant offset *of the network and $w(i)$ is called* weight of the addend x_i.

The bit vector function $pp : \mathbb{B}^n \times \mathbb{B}^m \to \mathbb{B}^{nm}$ *with*

$$pp(x_1, \ldots, x_n, y_1, \ldots, y_m) = (x_i \cdot y_j | i = 1, \ldots, n \text{ and } j = 1, \ldots, m)$$

is called ***partial product generator****.*

Every bit vector function $cmp : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ *with*

$$cmp(< x + k, n >, < y + k, n >) = cmp(< x, n >, < y, n >)$$

for all $k \in \mathbb{Z}$ *is called* ***comparator****.*

Partial product generators model bit-wise multiplication and comparators model comparison of bit vectors. Bit level addition units like **half adders (HA)** or **full adders (FA)** are modeled as addition networks. By construction, addition networks can be used to model any addition circuit ranging from *HAs* and *FAs* up to the entire addition scheme of a multiplier or a multiply-accumulate unit. This is true for both signed and unsigned arithmetic.

Example 5.3.2. *An signed 2×2-bit multiplier can be modeled with the partial product generator*

$$pp(x_0, x_1, y_0, y_1) = (x_0y_0, x_1y_0, x_0y_1, x_1y_1)$$

and the addition network

$$r(p_{0,0}, p_{1,0}, p_{0,1}, p_{1,1}) = \langle p_{0,0} - 2p_{1,0} - 2p_{0,1} + 4p_{1,1}, 4 \rangle$$

A simple bit level implementation of this multiplier may implement the addition network using a fulladder and two halfadders. They can be modeled by the addition networks $fa(a, b, c) =< a + b + c, 2 >$ and $ha(a, b) =< a + b, 2 >$, respectively.

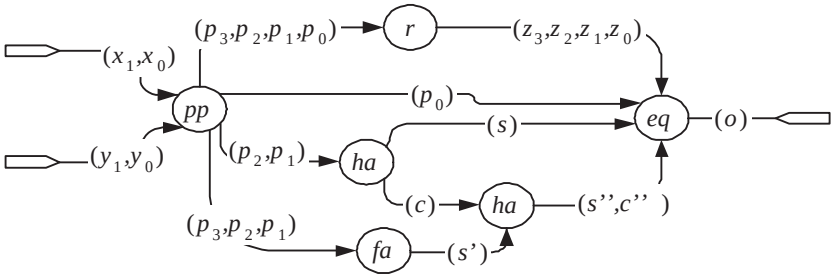


Figure 5.3.: ABL description for Example 5.3.2

For reasons of space we omit the formal definition of *ABL descriptions* as a DAG. The interested reader is referred to [87]. Basically, the nodes of the graph are labeled with their vertex type and the edges describe the interconnections between them. Here, we explain this concept by continuing Example 5.3.2.

Example 5.3.3. *The ABL description for the comparison of the bit level multiplier implementation discussed in Example 5.3.2 against its word level specification is depicted in Figure 5.3.*

The vertices of this graph are labeled with the bit vector function defined in the previous example. The edges (v, v') are labeled with bit vectors that propagate the result of v to the inputs of v'. In other words, the variables are defined by the following equations:

- $(p_0, p_1, p_2, p_3) = pp(x_0, x_1, y_0, y_1) = (x_0y_0, x_1y_0, x_0y_1, x_1y_1)$
- $(z_0, z_1, z_2, z_3) = r(p_0, p_1, p_2, p_4) = \langle p_0 - 2p_1 - 2p_2 + 4p_3, 4\rangle$
- $(s, c) = ha(p_1, p_2) = \langle p_1 + p_2, 2\rangle$
- $(s', c') = fa(p_1, p_2, p_3) = \langle p_1 + p_2 + p_3, 2\rangle$
- $(s'', c'') = ha(c, s') = \langle c + s', 2\rangle$
- $(o) = eq((z_0, z_1, z_2, z_3), (p_0, s, s'', c'')) = ((z_0, z_1, z_2, z_3) == (p_0, s, s'', c''))$

This example illustrates that ABL descriptions may contain structurally dissimilar representations for one and the same arithmetic function. To simplify the comparison of such representations a heuristic ad-hoc algorithm called *ABL normalization* was proposed in [87]. This algorithm performs a series of local equivalence transformations on the ABL description that are based on the commutative and distributive laws.

However, in the next section we will describe how to obtain a variety subset problem that is equivalent to the decision problem resulting from the comparison of such ABL representations. This paves the way for the application of generic computer algebra algorithms for which efficient implementations are available.

5.3.2. Using standard basis techniques to verify proof goals

Application of computer algebra techniques to ABL verification problems requires ABL components to be modeled by polynomials over a unique ring. Due to the operation mod used to specify ABL components, the ring $\mathbb{Z}/2^n$ seems to be the natural choice. However, the mapping of ABL descriptions on sets of polynomials $G \subset \mathbb{Z}/2^n[X]$ over such a ring is not trivial and will be detailed in this section. The key observation is that the constructed set G is a Gröbner basis of the generated ideal $I = \langle G\rangle$. This makes the proposed approach computational feasible.

We start with a set of equations $G_j, j = 1, \ldots, m$ given by polynomials $f_j \in \mathbb{Z}[X]$, X a finite set of variables, which are of the form

$$G_j : \sum_{i=0}^{n_j-1} 2^i r_i^{(j)} = f_j\left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)}\right) \mod 2^{n_j}.$$

For the variables $r_i^{(j)}, a_k^{(l)} \in X$ in this equation we assume $r_i^{(j)} \neq a_k^{(l)}$ for $1 \leq l \leq j$ and all i, k. We call the variables $a_i^{(j)}$ *inputs* and $r_i^{(j)}$ *outputs* of G_j.

Note that the equations G_j can be easily generated from the vertices of an ABL description and that the condition $r_i^{(j)} \neq a_k^{(l)}$ is fulfilled as the ABL description is acyclic by definition. For illustration we give a few examples.

Example 5.3.4. *The partial products of a non-Booth-encoded $n \times m$ multiplier can be modeled by the polynomial equations*

$$G_{i,k} : p_{i,k} = a_i b_k \ mod\ 2, (k = 0, \ldots, n-1, i = 0, \ldots, m-1)$$

Example 5.3.5. *A full adder with inputs a_0, a_1, a_2 and outputs s and c for sum and carry is modeled by the equation*

$$G_{FA} : 2c + s = a_0 + a_1 + a_2 \ mod\ 4$$

Example 5.3.6. *A k-bit adder with inputs $a = (a_0, \ldots, a_{k-1})$ and $b = (b_i)$ and result $r = (r_i)$ is modeled by*

$$G_{adder} : \sum_{i=0}^{k-1} 2^i r_i = \sum_{i=0}^{k-1} 2^i (a_i + b_i) \ mod\ 2^k$$

For every proof goal, we obtain an additional polynomial g depending on a subset of variables $\{a_1, \ldots, a_t\} \subset X$ and need to check whether

$$g(a_1, \ldots, a_t) = 0 \mod 2^n$$

for all solutions of the set of equations $\{G_j\}$.

Example 5.3.7. *A k-bit comparator of operands a and b is modeled by the polynomial*

$$g = \sum_{i=0}^{k-1} 2^i (a_i - b_i)$$

Denote the set of all solutions to $\{G_j\}$ as $V(\{G_j\})$. Analogously let $V(g)$ be the set of all roots of g. Usually the equations G_j and the polynomial g are given $\mod 2^k$ for different k. We apply a number of transformations to create an equivalent *variety subset problem* $V(\{h_i\}) \subset V(g)$ where h_i and g are polynomials over a single ring $\mathbb{Z}/2^N$ with appropriate N, which is necessary in order to apply computer algebra. To solve the problem we construct a Gröbner basis and then use normal form computations with respect to this basis.

For the reader's convenience we recall some basic facts about Gröbner basis theory (see also Chapter 1). We need a *monomial ordering* $<$, i.e., a well ordering on the set of monomials s.t. multiplication with a monomial respects the ordering. Here a *monomial* is a power product of variables and a term is the product of a monomial with a coefficient, i.e., an element of the ring $\mathbb{Z}/2^N$. Any polynomial $f \neq 0$ can be written as a finite sum of terms, $f = c_1 m_1 + \cdots + c_r m_r$ with c_i coefficients $\neq 0$ and m_i monomials s.t. $m_1 > m_2 > \cdots > m_r$. The largest term plays a special role and we call

$\mathrm{LM}(f) := m_1$ resp. $\mathrm{LC}(f) := c_1$ resp. $\mathrm{LT}(f) := c_1 m_1$ the *leading monomial* resp. the *leading coefficient* resp. the *leading term* of f.

Let $G \subset \mathbb{Z}/2^N[X]$ be a finite set of polynomials and $f \in \mathbb{Z}/2^N[X]$. If cm is any (non-zero) term of f and if cm is divisible by the leading term of an element $h \in G$ we say that f is reducible to $f' := f - (cm/\mathrm{LT}(h)) \cdot h$ and write $f \underset{h}{\rightarrow} f'$. The transitive and reflexive closure of the relation $\underset{h}{\rightarrow}$ is denoted by $\overset{*}{\underset{G}{\rightarrow}}$. If $f \overset{*}{\underset{G}{\rightarrow}} g$ and if g is not reducible by any h of G we call g a *normal form* of f w.r.t. G. This notion is, however, only useful if G is a Gröbner basis. In order to define a Gröbner basis we need the ideal $I = \langle G \rangle := \{\sum_{h \in G} f_h h | f_h \in \mathbb{Z}/2^N[X]\}$ generated by an arbitrary set G of polynomials. Note that for the set of solutions we have $V(I) = V(G)$ for any set of generators G. A set of generators G is called a (strong) Gröbner basis (of I) if $f \overset{*}{\underset{G}{\rightarrow}} 0$ for all $f \in I$. If G is a Gröbner basis then the normal form of any element $g \in \mathbb{Z}/2^N[X]$ is essentially unique and equal to 0 if and only if $f \in \langle G \rangle$.

5.3.3. Problem formulation over a single ring

Instead of directly converting the equations G_j into a set of polynomials over a single ring, we generate some additional equations. These equations are redundant in the sense that they can be derived from the original equations G_j. However, they will play an important role for the efficiency of the solution techniques described in Section 5.3.4. More precisely, these equations ensure that the polynomial system generated from them is a Gröbner basis of the corresponding ideal. This will be discussed later.

For every G_j we generate n_j equations

$$G_j^{(t)} : \sum_{i=0}^{t-1} 2^i r_i^{(j)} = f_j^{(t)} \left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)} \right) \mod 2^t$$

with $t = 1, \ldots, n_j$ and with $f_j^{(t)} = f_j \mod 2^t$ being the minimal polynomial (see Proposition 4.3.6) representing the same polynomial function $(\mathbb{Z}/2^t)^{m_j} \rightarrow \mathbb{Z}/2^t$ as f_j.

Obviously, every solution of the G_j is also a solution of the system $\{G_j^{(t)} \mid t = 1, \ldots, n_j\}$ and vice versa.

Let S be the set of variables (signals) occurring in g saturated with respect to the property that if $r_{t-1}^{(j)} \in S$ then all variables of $G_j^{(t)}$ are also in S. For the further course of action only the equations $G_j^{(t)}$ with $r_{t-1}^{(j)} \in S$ are relevant. The solution set for the variables in S does not change when omitting the other equations. Note that this corresponds to a cone-of-influence reduction on the netlist of a circuit.

Example 5.3.8. *Suppose the n bit final adder of a multiply/accumulate unit is reused for computation of an m-bit addition ($m < n$). In a property checking instance for this addition only the lowermost m bits of the adder take influence on the arithmetic result. By the above construction we only instantiate the equations*

$$G_{adder}^{(t)} : \sum_{i=0}^{t-1} 2^i r_i = \sum_{i=0}^{t-1} 2^i (a_i + b_i) \ mod\ 2^t$$

for $t < m$.

So far the equations $G_j^{(t)}$ use the operation $\bmod 2^t$ for different t, that is, we work over different rings $\mathbb{Z}/2^t$ and none is contained in the other (we have only surjections of rings $\mathbb{Z} \twoheadrightarrow \mathbb{Z}/2^{t'} \twoheadrightarrow \mathbb{Z}/2^t$ if $t' \geq t$). In order to apply Gröbner basis techniques to the problem we need to generate a set of polynomials over a single ring.

Let $N := n + \max\{n_j \mid j = 1, \ldots, m\}$ with n_k, n, m as above. We want to transform every equation into an element of the polynomial ring over $\mathbb{Z}/2^N$. To achieve this, we introduce new variables $s_t^{(j)}$ (called *slack variables*) and consider the polynomials

$$\widetilde{G}_j^{(t)} := \sum_{i=0}^{t-1} 2^i r_i^{(j)} - f_j^{(t)}\left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)}\right) - 2^t s_t^{(j)}.$$

The set of common roots for the $\widetilde{G}_j^{(t)}$ projected on the variables in S corresponds to $V(\{G_j\})$. We can omit some of the extra variables $s_t^{(j)}$ if we know that $0 \leq f_j^{(t)} \leq 2^t - 1$ holds over $\mathbb{Z}$. If this condition cannot be guaranteed and we need to know the exact value of $s_t^{(j)}$ during the computation we can replace $s_t^{(j)}$ by a polynomial in the variables $a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)}$, i.e., a subset of the inputs of G_j. For example, the polynomial modeling a half adder $r_0 - a_0 - a_1 + 2s$ results in the polynomial $s = a_0 a_1$ for the slack variable. However, often it is better to introduce the slack variables because, in general, the polynomials for the slack variables will be very large even for small polynomials $f_j^{(t)}$.

Let $G = \{\widetilde{G}_j^{(t)} \mid j = 1, \ldots, m \text{ and } t = 1, \ldots, n_j\}$ and $I = \langle G \rangle$ be the ideal generated by this set. Using the language of computer algebra the decision problem can be formulated by the following question:
Is $V(I) \subset V(2^{N-n}g)$, where $V(I)$ and $V(f)$ denote the set of all common roots (in $(\mathbb{Z}/2^N)^k$, where k is the number of variables) of the polynomials in I and the set of roots of the polynomial $2^{N-n}g$, respectively?

In the next section we will detail how to efficiently solve this problem.

5.3.4. Solving decision problems at the ABL

The following proposition turns out to be the key for an effective solution of the presented problem.

Proposition 5.3.9. *The set $G = \{\widetilde{G}_j^{(t)}\}$ is a Gröbner basis with respect to any monomial ordering refining the following partial ordering*

$$r_i^{(j)} > \textit{every monomial in the variables } a_k^{(j)}, s_t^{(j)}, r_l^{(j)}$$

for all i, k, t, j and $l < i$.

Proof. Let $<$ be a monomial ordering as required in the statement. We need to show that it is not possible to generate a polynomial from the polynomials in G with a leading term that is not divisible by any leading term of the polynomials in G. It is sufficient to show (see Corollary 2.2.10):

(1) For any two polynomials $f, g \in G$ the normal form of

$$\frac{\operatorname{lcm}(\operatorname{LT}(f), \operatorname{LT}(g))}{\operatorname{LT}(f)} f - \frac{\operatorname{lcm}(\operatorname{LT}(f), \operatorname{LT}(g))}{\operatorname{LT}(g)} g$$

with respect to G is zero.

(2) For any $f \in G$ the normal form of $\frac{2^N}{\operatorname{LC}(f)} f$ is zero.

A slight generalization of the strong product criterion (see Lemma 2.2.11) states that (1) is fulfilled, as the polynomials have different variables in their leading terms and these variables do not occur in any other term of the corresponding polynomials. Now let $f = G_j^{(t)}$. We obtain

$$\begin{aligned}
\operatorname{LT}\left(\frac{2^N}{\operatorname{LC}(f)} f\right) &= \operatorname{LT}\Big(\underbrace{2^{N-t+1} G_j^{(t)}}_{\|}\Big) \\
&= \operatorname{LT}\Big(\overbrace{2^{N-t+1} G_j^{(t-1)}}\Big) \\
&= 2^{N-t+1} \operatorname{LT}\left(G_j^{(t-1)}\right)
\end{aligned}$$

as $2^{N-t+1} \cdot 2^{t-1} r_{t-1}^{(j)} = 0 = 2^{N-t+1} \cdot 2^{t-1} s_{t-1}^{(j)}$ and $2^{N-t+1} \cdot 2^{t-2} r_{t-2}^{(j)} \neq 0$, and since the polynomials $f_j^{(t)}$ appearing in $\widetilde{G}_j^{(t)}$ are chosen minimal. In the first step of the normal form algorithm we will select $G_j^{(t-1)}$ and reduce $2^{N-t+1} G_j^{(t)}$ to zero. This shows (2). □

By Lemma 5.3.10 we prove that normal form computation can be used as an effective solution procedure for the problem at hand.

Lemma 5.3.10. *Let G be a Gröbner basis of an ideal $I \subset \mathbb{Z}/2^N[\mathbf{x}]$, $\mathbf{x} = (\mathbf{x}', \mathbf{x}'')$, and g a polynomial such that h, the normal form of g w.r.t. G, is in $\mathbb{Z}/2^N[\mathbf{x}']$. Assume that for all $\mathbf{x}'$ there exist $\mathbf{x}''$ with $f(\mathbf{x}', \mathbf{x}'') = 0$ for all $f \in G$. Then h defines the zero function if and only if $V(G) \subset V(g)$.*

Proof. If h defines the constant zero function the set $V(h) \cap V(G) = V(g) \cap V(G)$ contains all points and therefore $V(G) \subset V(g)$ is trivial. Assume that for the variables $\mathbf{x}'$ of h a valuation exists such that h is not zero. By assumption we can extend this valuation to a valuation on all variables such that $f(\mathbf{x}) = 0, f \in G$. It follows $V(G) \not\subset V(g)$. □

Let $g \in \mathbb{Z}/2^n[\mathbf{x}]$ and h be the normal form of $2^{N-n} g$ with respect to G, which can be computed [9] by Algorithm 10. Since we are only interested in the function of h on $V(I)$ we can always replace portions of h by equivalent polynomials with respect to $V(I)$. In particular, we can replace every slack variable in the normal form by a polynomial expression in the inputs of the corresponding equation G_j. Therefore we may assume that h does not contain any slack variables. Furthermore, the output variables of the equations G_j do not occur in h as otherwise h would be reducible by some of the generated sub-identities $G_j^{(t)}$, hence h satisfies the assumptions of Lemma 5.3.10.

This guarantees that the variables present in h are inputs to the ABL description. Every valuation of these variables can be extended to a consistent valuation for the signals of the ABL. Further we can effectively decide whether h defines the zero function for all rings $\mathbb{Z}/m$ (see Chapter 4) and therefore decide the ABL problem by Lemma 5.3.10.

Algorithm 10 Normal form algorithm

Input: f a polynomial, G a finite set of polynomials,
$>$ a monomial ordering
Output: A normal form of f
while $f \neq 0$ and $\emptyset \neq G' = \{g \in G \ : \ \mathrm{LT}(g) \mid \mathrm{LT}(f)\}$ **do**
Select $g \in G'$
Let $\mathrm{LT}(f) = m \cdot \mathrm{LT}(g)$ with $m \cdot \mathrm{LC}(g) \neq 0$
$f := f - m \cdot g_i$
end while
return f

As already noted in Section 5.3.3 it is not always efficient to replace all remaining slack variables by polynomial expressions in terms of the input variables of the corresponding equations. Therefore we use special procedures for the practical computations, which we do only hint in the next section.

5.3.5. Heuristics

At the end of the previous section we assumed that the normal form of the polynomial representing the proof goal does not contain any slack variables. However, as already noted in Section 5.3.3 it is not always efficient to replace all remaining slack variables by polynomial expressions in terms of the input variables of the corresponding equations. In order to cope with this problem we suggest two special treatment procedures we used for our implementation.

Assume that we have computed the normal form h of $2^{N-n}g$ as presented in Section 5.3.4. First consider the case that h contains several slack variables originating from the same original equation G_j, i.e., the slack variables result from different subidentities $G_j^{(t)}$ of the equation G_j. In this case we can select one of these slack variables s and express all other slack variables by polynomials in terms of s and input variables of G_j.

Practical experience shows that these polynomials usually remain of manageable size in contrast to the polynomials obtained by replacing all slack variables with expressions in the inputs of a block. To give the reader an intuition we would like to illustrate this issue by means of a little example.

Example 5.3.11. *Let*

$$\widetilde{G}_{adder}^{(t)} = \sum_{i=0}^{t-1} 2^i r_i - \sum_{i=0}^{t-1} 2^i (a_i + b_i) - s_t.$$

We know that $2s_t + r_{t-1} = a_{r-1} + b_{r-1} + s_{t-1}$. *In case of an addition not only the variables* a_i *and* b_i *are restricted to* $\{0,1\}$ *but also the slack variables. This yields the short expression of* s_t *in terms of* s_{t-1},

$$s_t = a_{r-1}b_{r-1} + a_{r-1}s_{t-1} + b_{r-1}s_{t-1} - 2a_{r-1}b_{r-1}s_{t-1},$$

while writing s_t *only with the variables* a_i, b_i *produces a huge polynomial.*

To conclude the treatment of slack variables we now focus on the case that the remaining slack variables originate from different original equations G_j. In practice we often encounter the case that $h = 2^m s_t^{(j)} + 2^m f$. We can deduce that $2^{N-n}g$ is zero modulo 2^k for all $k \leq m$. Based on h we calculate a polynomial $\tilde{h}$ by substituting $s_t^{(j)}$ by the polynomial $f_j^{(t)}\left(a_1^{(j)}, a_2^{(j)}, \ldots, a_{m_j}^{(j)}\right)$. We know the value of $s_t^{(j)}$ is determined by the slice of the value $f_j^{(t)}$ which exceeds $2^m - 1$. This implies that in order to prove $h = 0$ and hence $g = 0$ it is sufficient to show $\tilde{h} - \tilde{h} \mod 2^m = 0$ for all possible values of $\tilde{h}$.

Experimental experience demonstrates that $\tilde{h} - \tilde{h} \mod 2^m = 0$ can be proven effectively with an additional normal form computation. Note that this especially applies to all the industrial examples presented in the experimental results section.

5.4. Experimental results

In order to evaluate the techniques presented in the previous sections we conducted a series of experiments. Except for one experiment explicitly indicated in the sequel, all experiments were carried out on a machine running Suse Linux 10.3 on a Intel Core 2 Duo E6400 with 8 GB RAM.

The algorithms presented in Section 5.3.2 have been implemented within the framework of the general purpose computer algebra system Singular [35]. We used the industrial formal property checker Onespin 360 MV [59] to generate bit vector netlists for the considered verification problems. From these bit-vector netlists we extracted an arithmetic bit level description for the arithmetic parts of the decision problem and dumped out the resulting ABL description. The resulting problem file is used to generate the variety subset problem that is handed over to Singular in order to find a solution.

As a first step of the evaluation we used a number of parameterized benchmarks to evaluate the scalability of the proposed approach with respect to the bit-width of the datapath under verification. The benchmark suite consists of two instances (distrib and commute) for word level implementations of the functions $ab + ac$ and $(ab)c$ where commutative and distributive laws have been applied to the word level operands, a bit level implementation of an unsigned multiplier with Booth-encoded partial products (mult_ub) and a sequential implementation for the multiplication of four values with a single multiplier (shared).

We compare the performance in terms of run-time of the presented solution based on Singular against the normalization approach of [87], a SAT-based decision procedure based on bit blasting, and the SMT solver Spear v.2.0 for the theory of fixed-size

bit-vector functions (QF-BV). Note that an earlier version of Spear showed the best performance in this category on the 2007 SMT competition.

Instance	**Bit-Width**	**Normalizer**	**SAT**	**SMT**	**Singular**
distrib	4	0,01	0,28	0,40	0,69
distrib	8	0,03	> 3600s	> 3600s	0,66
distrib	16	0,10	> 3600s	> 3600s	0,97
distrib	32	0,81	> 3600s	> 3600s	2,19
distrib	64	14,33	> 3600s	> 3600s	11,30
commute	4	0,02	0,55	1,01	0,69
commute	8	0,08	> 3600s	> 3600s	0,67
commute	16	1,40	> 3600s	> 3600s	1,09
commute	32	57,17	> 3600s	> 3600s	3,56
commute	64	2794,67	> 3600s	> 3600s	26,03
mult_ub	4	0,02	0,02	0,15	0,66
mult_ub	8	0,13	41,53	> 3600s	0,96
mult_ub	16	2,21	> 3600s	> 3600s	3,87
mult_ub	32	53,55	> 3600s	> 3600s	79,04
mult_ub	64	1136,14	> 3600s	> 3600s	> 8 GB
shared	4	0,04	2,83	16,78	0,97
shared	8	0,46	> 3600s	> 3600s	0,64
shared	16	39,79	> 3600s	> 3600s	1,09
shared	32	2707,72	> 3600s	> 3600s	20,25

Table 5.1.: CPU-times(s) of scalability experiments

Table 5.1 summarizes the results of these experiments. The table is organized as follows. Columns one and two contain instance and operand bit-width of the datapath. The remaining columns show the CPU times required by the particular tool to prove the instance. In case the memory limit or timeout limit was reached this is indicated by "> 8 GB" and "> 3600", respectively.

In order to evaluate the performance of Singular with respect to other computer algebra systems we also report results for solving the generated variety subset problems with the industrial computer algebra tool Magma [7]. However, due to license restrictions, these results were obtained using another machine, namely an AMD Dual Opteron 2.2 GHz with 16 GB RAM running Linux. We re-ran the Singular problems on this machine in order to allow for comparison of the run times. For the comparison we also increased the memory limit to 16GB. Table 5.2 summarizes the results for this comparison.

The presented results of the scalability experiments indicate that the proposed modeling and the proposed algorithms are adequate to solve verification problems with industrial impact. To demonstrate this we investigated a property suite originating from the verification of Infineon's Tricore 2 processor. The processor has advanced DSP features including a sophisticated integer pipeline that provides a large variety of multiply

Instance	Bit-Width	Singular	Magma
distrib	16	1,08	2,33
distrib	32	2,70	15,61
distrib	64	14,53	> 16 GB
commute	16	1,35	5,53
commute	32	4,71	46,07
commute	64	38,96	> 16 GB
mult_ub	4	0,56	> 3600s
mult_ub	16	4,07	> 3600s
mult_ub	32	85,77	> 3600s
shared	4	0,46	1,08
shared	8	0,66	1,35
shared	16	1,37	3,09
shared	32	32,27	108,01

Table 5.2.: CPU-times(s) of scalability experiments

and multiply/accumulate instructions. The properties in the investigated property suite verify that every variant of these instructions causes the integer pipeline of the processor to deliver the expected arithmetic result according to the architectural manual. In order to obtain a high degree of resource sharing large portions of the datapath have been implemented at the arithmetic bit level and sophisticated control logic is used for configuration according to the executed instructions.

We used the techniques of [87] to generate the decision problems at the arithmetic bit level. All the resulting decision problems could be solved with Singular when modeled by polynomials as presented in this document. Table 5.3 shows the results for a representative subset of the problem instances derived from the Tricore 2 property suite. It is organized as follows. The first column shows the commitment of the property specifying the arithmetic result of the integer instruction under verification. Columns two and three show the run-time of the normalization approach and the corresponding Singular run-time. Unless explicitly indicated all operations are considered as signed operations on the specified bit-vectors.

In essence, all experiments show that the presented approach outperforms the ad-hoc normalization approach in terms of CPU time. Moreover, algorithms and modeling rely on a well-understood mathematical foundation which opens ample opportunities for further extensions of this framework.

However, the use of a generic computer algebra system as Singular for solving the normalization problems is paid with a price in terms of memory consumption. Except for some of the problems where the ABL description is generated from word level problems Singular typically requires 3–8 GB of memory. This is caused by the data structures used inside Singular to represent polynomials.

These data structures are not optimized with respect to the characteristics of the

Datapath result	Normalizer	Singular
res[31:0]=op3[31:0]+(op1[31:0]*op2[31:0]);	49,94	4,42
res[31:0]=op3[31:0]+(op1[15:0]*op2[31:16]<<1); res[63:32]=op3[63:32]+(op1[31:16]*op2[15:0]<<1);	39,72	2,28
res[31:0]=op3[31:0]+(op1[31:16]*op2[31:16]<<1);	18,39	2,47
res[15:0]=rnd16(op3[31:0]+(7FFFFFFF)); res[31:16]=rnd16(op3[63:32] +(op1[31:16]*op2[15:0]<<1));	19,90	2,46
res[63:0]=op3[63:0]+(op1[31:16]*op2[15:0]<<16) -(op1[31:16]*op2[31:16]<<16)	31,04	8,77
res[63:0]=op3[31:0]-(op1[31:0]*op2[31:0]);	55,19	20,01
res[63:0]=op3[63:0]-(op1[31:16]*op2[15:0]<<16) -(op1[15:0]*op2[15:0]<<16)	27,18	9,64
res[63:0]=op1[31:0]*op2[31:0]; (unsigned)	57,33	14,73
res[31:16]=rnd16(op1[31:16]*op2[31:16]);	17,41	2,21

Table 5.3.: Results for selected Tricore 2 properties

problems considered here. Compared to problems typically considered in computer algebra, we consider a large number of variables, and many polynomials. On the other hand the individual polynomials have low degree and only use a small fraction of the variables. With application-specific implementations of the employed algorithms such as the normal form computation a great improvement of the memory efficiency can be obtained. Such an implementation was created and is maintained by Alexander Dreyer (see [61]) after the publication of the articles corresponding to this chapter.

5.5. Conclusion

Decision problems at the arithmetic bit level have been modeled using polynomials over rings $\mathbb{Z}/2^n$. It has been proven that the generated sets of polynomials form a Gröbner basis with respect to certain monomial orderings that can easily be determined using the topological ordering of design signals. This allows for utilization of the normal form algorithm to efficiently solve a variety subset problem that is equivalent to the original decision problem.

By this means we provide a solid mathematical foundation to the ad-hoc technique of arithmetic bit level normalization. The developed techniques have proven to be applicable to verification problems of industrial size.

Part II.

Distributive lattices and rank tests

Introduction to Part II

Rank tests are statistical tests concerned with putting an order on different items. The need for an order may be manifold starting by comparing different countries with respect to several indicators (e. g. think of the many WHO or OECD studies) not ending with giving an indicator about the next genes to create markers for.

The techniques and algorithm presented in the next part were applied to time series of microarray data (see [25, 26, 56]) obtained during laborious experiments at the Stowers Institute for Medical Research in Kansas City:

Time series of microarray data Due to the microarray technique it is feasible to measure the expression of all known genes at several times in a biological process. Since the number of genes is very high (>20,000) a decision regarding the genes for further research has to be taken. Further research consists for example in producing specialized markers for certain genes which requires a lot of effort. Hence a ranking of the genes according to the likelihood that they are involved in the process under investigation is required. Rank tests give rise to such a ranking in a non-parametric way.

The data created by the Stowers Institute for Medical Research concerns processes and pathways regarding the development of vertebra within embryos. Therefore time series of 15 to 20 data points consisting of roughly 20,000 to 30,000 mRNA expression levels were generated. Our algorithms were able to analyze and rank the different mRNA entries and proved to be helpful for further analysis (see [24]).

The following four further examples are copied from "Ranking and Prioritization for Multi-Indicator Systems" (see [13]):

Chemicals assessment Chemicals as useful as they are, they can be harmful to humans and the environment. Therefore it appears rather clear that only those chemicals should be used in the market that do not have an adverse impact on humans and the environment. How do we find out whether they are hazardous? There are many time-consuming and expensive investigations necessary to perform a risk assessment. Hence the question is: with which chemicals to begin at first? Thus a ranking can be performed to give the more involved investigations a reasonable operating sequence. Once accepted that a ranking is needed, we discover that there is no intrinsic property of a chemical which tells us that it is hazardous. Still worse, one needs to know the hazard of chemicals in different scenarios. Hence, several aspects of a chemical need to be simultaneously considered. And thus the final and central question arises: how to rank chemicals characterized by several attributes.

Child well-being In a report of UNICEF, a ranking of twenty one rich nations was performed with respect to child well-being. For this purpose, six attributes were finally constructed by which the countries were ranked. It is clear that each of these six rankings need not be the same. Therefore, a composite indicator was defined, giving each of the six indicators the same weight. How far is this justified? What influence does this kind of aggregation have on the final result? Italy for example could get a better position, if the indicator "family" would get more weight on the index. How can we analyze the role of weights?

Integrity of watersheds Scientists of the Atlantic Slope Consortium, (ASC) developed three levels of indicators to describe the health of watersheds. The indicators of the three levels increase in quality and accuracy of the data as well as the amount of cost and efforts needed to obtain the data. An important question is about how well level one or level two indicators perform compared with level three indicators. Partial order can help with this question.

Surface waters management strategies High concentrations of nutrients such as Phosphorus or Nitrogen in surface waters is of much concern for environmental protection agencies. What could be done to improve the situation? Clearly one has to study the release paths by which nutrients enter the surface waters. Then one has to develop strategies to control these and limit the emissions into surface waters. How well do such strategies work?

The ranking algorithms presented in the next chapters are based on computing the number of linear extensions for one or many partial orders. The problem is well known in computer science to be #P-complete [10] and optimal algorithms for listing the linear extensions have been devised (see [67]). However, it is possible to compute the number of linear extensions required for many applications magnitudes faster by using a completion based method on distributive lattice developed by the author (see Chapter 8). The corresponding algorithm is able to compute the previously unknown number $|L(2^{[5]})| = 14,807,804,035,657,359,360$ (see [82]) in less than a second. This is the number of linear extensions of the partial order on the set of all subsets of $\{1,2,3,4,5\}$ given by the inclusion relation. Further the number corresponds to all linear orderings of the vertices of the 5-cube containing the partial order induced by the component-wise natural order.

Chapter 6.

Geometry of rank tests

Convex rank tests are partitions of the symmetric group which have desirable geometric properties. The statistical tests defined by such partitions involve counting all permutations in the equivalence classes. Each class consists of the linear extensions of a partially ordered set specified by data. The methods refine existing rank tests of non-parametric statistics, such as the sign test and the runs test, and are useful for exploratory analysis of ordinal data. We establish a bijection between convex rank tests and probabilistic conditional independence structures known as semigraphoids. The subclass of submodular rank tests is derived from faces of the cone of submodular functions, or from Minkowski summands of the permutohedron. We enumerate all small instances of such rank tests. Of particular interest are graphical tests, which correspond to both graphical models and to graph associahedra.

This chapter is published as "Convex rank tests and semigraphoids" [55] in the SIAM Journal on Discrete Mathematics together with Jason Morton, Lior Pachter, Anne Shiu and Bernd Sturmfels. This paper itself is an expanded version of the note "Geometry of Rank Tests" [54].

The research on rank tests originated in discussions with Olivier Pourquié and Mary-Lee Dequéant as part of the DARPA Program *Fundamental Laws of Biology*, that supported Jason Morton, Lior Pachter, and Bernd Sturmfels. Anne Shiu was supported by a Lucent Technologies Bell Labs Graduate Research Fellowship, and Oliver Wienand by the Wipprecht foundation. Milan Studený and František Matúš provided help- and insightful comments.

6.1. Introduction

The non-parametric approach to statistics was introduced by Pitman [64] via the method of permutation testing. Subsequent development of these ideas revealed a close connection between non-parametric tests and rank tests, which are statistical tests suitable for ordinal data. Beginning in the 1950s, many rank tests were developed for specific applications, such as the comparison of populations or testing hypotheses for determining the location of a population. The geometry of these tests was explored in [20]. More recently, the search for patterns in large datasets has spurred the development and exploration of new tests. For instance, the emergence of microarray data in molecular biology has led to tests for identifying significant patterns in gene expression time series; see e.g. [90]. This

application motivated us to develop a mathematical theory of rank tests. We propose that a **rank test** is a partition of S_n induced by a map $\tau : S_n \to T$ from the symmetric group of all permutations of $[n] = \{1, \ldots, n\}$ onto a set T of statistics. The statistic $\tau(\pi)$ is the *signature* of the permutation $\pi \in S_n$. Each rank test defines a partition of S_n into classes, where π and π' are in the same class if and only if $\tau(\pi) = \tau(\pi')$. We identify $T = \text{image}(\tau)$ with the set of all classes in this partition of S_n. Assuming the uniform distribution on S_n, the probability of seeing a particular signature $t \in T$ is $1/n!$ times $|\tau^{-1}(t)|$. The computation of a p-value for a given permutation $\pi \in S_n$ leads to the problem of summing

$$\Pr(\pi') \quad = \quad \frac{1}{n!} \cdot |\, \tau^{-1}\big(\tau(\pi')\big)\,| \tag{6.1}$$

over permutations π' with $\Pr(\pi') \leq \Pr(\pi)$, a computational task to be addressed in Chapter 8.

The emphasis of the following discussion is on the mathematics underlying rank tests, and, in particular, on the connection to statistical learning theory (semigraphoids). We refer to [56] for details on how to use the presented rank tests in practice, and how to interpret the p-values derived from (6.1).

The five subsequent sections are organized as follows. In Section 2 we explain how existing rank tests in non-parametric statistics can be understood from the presented geometric point of view, and how they are described in the language of algebraic combinatorics [77]. In Section 3 we define the class of *convex rank tests*. These tests are most natural from both the statistical and the combinatorial point of view. Convex rank tests can be defined as polyhedral fans that coarsen the hyperplane arrangement of S_n. The main result (Theorem 6.3.3) states that convex rank tests are in bijection with conditional independence structures known as *semigraphoids* [21, 63, 80].

Section 4 is devoted to convex rank tests that are induced by submodular functions. These *submodular rank tests* are in bijection with Minkowski summands of the $(n-1)$-dimensional permutohedron and with *structural imset models*. These tests are at a suitable level of generality for the biological applications [56, 90] that motivated us. The connection between polytopes and independence models is made concrete in the classification of small models in Remarks 6.4.6–6.4.8.

In Section 5 we study the subclass of *graphical tests*. In combinatorics, these correspond to graph associahedra, and in statistics to graphical models. The equivalence of these two structures is shown in Theorem 6.5.2. The implementation of convex rank tests requires the efficient enumeration of linear extensions of partially ordered sets. The algorithms and software are discussed in Chapter 8. A key ingredient is the efficient computation of distributive lattices.

6.2. Rank tests and posets

A permutation π in S_n is a total order on the set $[n] := \{1, \ldots, n\}$. This means that π is a set of $\binom{n}{2}$ ordered pairs of elements in $[n]$. For example, $\pi = \{(1,2), (2,3), (1,3)\}$

represents the total order $1 > 2 > 3$. If π and π' are permutations then $\pi \cap \pi'$ is a partial order.

In the applications we have in mind, the data are vectors $u \in \mathbb{R}^n$ with distinct coordinates. The permutation associated with u is the total order $\pi = \{\, (i,j) \in [n] \times [n] \,:\, u_i < u_j \,\}$. We shall employ two other ways of writing a permutation. The first is the **rank vector** $\rho = (\rho_1, \ldots, \rho_n)$, whose defining properties are $\{\rho_1, \ldots, \rho_n\} = [n]$ and $\rho_i < \rho_j$ if and only if $u_i < u_j$. That is, the coordinate of the rank vector with value i is at the same position as the ith smallest coordinate of u. The second is the **descent vector** $\delta = (\delta_1 | \delta_2 | \ldots | \delta_n)$. The descent vector is defined by $u_{\delta_i} > u_{\delta_{i+1}}$ for $i = 1, 2, \ldots, n-1$. Thus the ith coordinate of the descent vector is the position of the ith largest value of the data vector u. For example, if $u = (11, 7, 13)$ then its permutation is represented by $\pi = \{(2,1), (1,3), (2,3)\}$, by $\rho = (2,1,3)$, or by $\delta = (3|1|2)$.

A permutation π is a **linear extension** of a partial order P on $[n]$ if $P \subseteq \pi$, i.e. π is a total order that refines the partial order P. We write $\mathcal{L}(P) \subseteq S_n$ for the set of linear extensions of P. A partition τ of the symmetric group S_n is a **pre-convex rank test** if the following axiom holds:

$$\text{(PC)} \qquad \text{If } \tau(\pi) = \tau(\pi') \text{ and } \pi'' \in \mathcal{L}(\pi \cap \pi') \text{ then } \tau(\pi) = \tau(\pi') = \tau(\pi'').$$

Note that $\pi'' \in \mathcal{L}(\pi \cap \pi')$ means $\pi \cap \pi' \subseteq \pi''$. The number of all rank tests τ on $[n]$ is the **Bell number** $B_{n!}$, which is the number of set partitions of a set of cardinality $n!$.

Example 6.2.1. *For $n = 3$ there are $B_6 = 203$ rank tests, or partitions of the symmetric group S_3, which consists of six permutations. Of these 203 rank tests, only 40 satisfy the axiom (PC). One example is the pre-convex rank test in Figure 1. Here the symmetric group S_3 is partitioned into the four classes $\{(1|2|3)\}$, $\{(2|1|3)\}$, $\{(2|3|1)\}$, and $\{(1|3|2), (3|1|2), (3|2|1)\}$.*

Each class C of a pre-convex rank test τ corresponds to a poset P on the ground set $[n]$; namely, the partial order P is the intersection of all total orders in that class: $P = \bigcap_{\pi \in C} \pi$. The axiom (PC) ensures that C coincides with the set $\mathcal{L}(P)$ of all linear extensions of P. The inclusion $C \subseteq \mathcal{L}(P)$ is clear. The proof of the reverse inclusion $\mathcal{L}(P) \subseteq C$ is based on the fact that, from any permutation π in $\mathcal{L}(P)$, we can obtain any other π' in $\mathcal{L}(P)$ by a sequence of reversals $(a,b) \mapsto (b,a)$, where each intermediate $\hat{\pi}$ is also in $\mathcal{L}(P)$. Consider any $\pi_0 \in \mathcal{L}(P)$ and suppose that $\pi_1 \in C$ differs by only one reversal $(a,b) \in \pi_0$, $(b,a) \in \pi_1$. Then $(b,a) \notin P$, so there is some $\pi_2 \in C$ such that $(a,b) \in \pi_2$; thus, $\pi_0 \in \mathcal{L}(\pi_1 \cap \pi_2)$ by (PC). This shows $\pi_0 \in C$.

A pre-convex rank test therefore can be characterized by an unordered collection of posets $P_1, P_2, \ldots, P_k$ on $[n]$ that satisfies the property that the symmetric group S_n is the disjoint union of the subsets $\mathcal{L}(P_1), \mathcal{L}(P_2), \ldots, \mathcal{L}(P_k)$. This structure was discovered independently and studied by Postnikov, Reiner and Williams [66, §3] who used the term *complete fan of posets* for what we shall call a convex rank test in Section 3. The posets $P_1, P_2, \ldots, P_k$ that represent the classes in a pre-convex rank test capture the shapes of data vectors. In graphical rank tests (Section 6.5), this shape can be interpreted as a smoothed topographic map of the data vector.

Example 6.2.2 (The sign test for paired data)**.** *The* ***sign test*** *is performed on data that are paired as two vectors* $u = (u_1, u_2, \ldots, u_m)$ *and* $v = (v_1, v_2, \ldots, v_m)$*. The null hypothesis is that the median of the differences* $u_i - v_i$ *is 0. The test statistic is the number of differences that are positive. This test is a rank test, because* u *and* v *can be transformed into the overall ranks of the* $n = 2m$ *values, and the rank vector entries can then be compared. This test coarsens the convex rank test which is the MSS test of Section 4 with* $\mathcal{K} = \{\{1, m+1\}, \{2, m+2\}, \ldots\}$*.*

Example 6.2.3 (Runs tests)**.** *A* runs test *can be used when there is a natural ordering on the data points, such as in a time series. The data are transformed into a sequence of 'pluses' and 'minuses,' and the null hypothesis is that the number of observed runs is no more than that expected by chance. Common types of runs tests include the sequential runs test ('plus' if consecutive data points increase, 'minus' if they decrease), and the runs test to check randomness of residuals, i.e. deviation from a curve fit to the data. A runs test is a coarsening of a convex rank test, known as* ***up-down analysis*** *[90, §6.1.1], which is described in Example 6.3.4 below.*

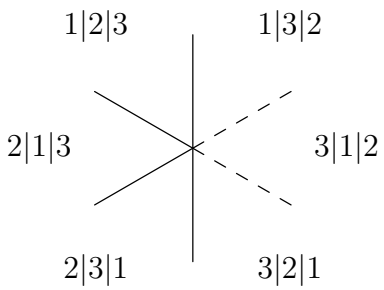


Figure 6.1.: Illustration of a pre-convex rank test that is not convex. Cones are labeled by descent vectors, so 1|2|3 indicates the cone $u_1 > u_2 > u_3$. This rank test is specified by the four posets $P_1 = \{3{<}1, 2{<}1, 3{<}2\}$, $P_2 = \{1{<}2, 3{<}2, 3{<}1\}$, $P_3 = \{3{<}2, 1{<}3, 1{<}2\}$ and $P_4 = \{2{<}3\}$.

These two examples suggest that many rank tests from classical non-parametric statistics have a natural refinement by a pre-convex rank test. However, not all tests have this property. Because many classical rank tests apply to loosely grouped data (e.g. data which are divided into two samples), the axiom (PC) is not always satisfied. In such cases, the pre-convex rank test is a first step, after which permutations are grouped together under additional symmetries, e.g., the permutations $\delta = (1|2|3|4|5)$ and $\delta' = (5|4|3|2|1)$ might be identified.

The adjective "pre-convex" refers to the following interpretation of the axiom (PC). Consider any two data vectors u and u' in $\mathbb{R}^n$, and a convex combination $u'' = \lambda u + (1 - \lambda)u'$, with $0 < \lambda < 1$. If π, π', π'' are the permutations of u, u', u'' then $\pi'' \in \mathcal{L}(\pi \cap \pi')$. Thus the equivalence classes in $\mathbb{R}^n$ specified by a pre-convex rank test are convex cones. In the next section, we shall remove the prefix from "pre-convex" if the faces of these cones fit together well.

6.3. Convex rank tests

A *fan* in $\mathbb{R}^n$ is a finite collection $\mathcal{F}$ of polyhedral cones [92] which satisfies the following properties:

(i) if $C \in \mathcal{F}$ and C' is a face of C, then $C' \in \mathcal{F}$,

(ii) if $C, C' \in \mathcal{F}$, then $C \cap C'$ is a face of C.

Two vectors u and v in $\mathbb{R}^n$ are **permutation equivalent** when $u_i < u_j$ if and only if $v_i < v_j$, and $u_i = u_j$ if and only if $v_i = v_j$ for all $i, j \in [n]$. Note that for two data vectors, each with distinct coordinates, they are permutation equivalent if and only if they have the same rank vector. The permutation equivalence classes (of which there are 13 for $n = 3$) induce a fan called the *S_n-fan*. The arrangement of hyperplanes $\{x_i = x_j\}$ that defines these classes is also known as the **braid arrangement**, and its regions as the **Weyl chambers** of the Lie algebra $\mathfrak{sl}(n)$. The maximal cones in the S_n-fan, which are the closures of the permutation equivalence classes, are indexed by permutations δ in S_n. A **coarsening** of the S_n-fan is a fan $\mathcal{F}$ such that each permutation equivalence class of $\mathbb{R}^n$ is fully contained in a cone C of $\mathcal{F}$. Such a fan $\mathcal{F}$ defines a partition of S_n because each maximal cone of the S_n-fan is contained in some cone $C \in \mathcal{F}$.

Definition 6.3.1. *A **convex rank test** is a partition of the symmetric group S_n which is induced by a coarsening of the S_n-fan. We identify the fan with that rank test.*

We say that two maximal cones, indexed by δ and δ', of the S_n-fan *share a wall* if there exists an index k such that $\delta_k = \delta'_{k+1}$, $\delta_{k+1} = \delta'_k$, and $\delta_i = \delta'_i$ for $i \notin \{k, k+1\}$. This condition means that the corresponding permutations δ and δ' differ by an adjacent transposition. To such an unordered pair $\{\delta, \delta'\}$, we associate the following *(elementary) conditional independence (CI) statement*:

$$\delta_k \perp\!\!\!\perp \delta_{k+1} \,|\, \{\delta_1, \ldots, \delta_{k-1}\}. \tag{6.2}$$

The notation was coined by Dawid [21], where it is used to formally describe conditional independence among sets of random variables; we will see the connection shortly. For $k = 1$ we use the standard convention to abbreviate $\delta_1 \perp\!\!\!\perp \delta_2 \,|\, \{\,\}$ by $\delta_1 \perp\!\!\!\perp \delta_2$.

Example 6.3.2. *For $n = 3$ there are 40 pre-convex rank tests (Example 6.2.1), but only 22 of them are convex rank tests. The corresponding CI models are shown in Figure 5.6 on page 108 in [80].*

The formula (6.2) defines a map from the set of walls of the S_n-fan onto the set

$$\mathcal{T}_n \;:=\; \big\{\, i \perp\!\!\!\perp j \,|\, K \,:\, K \subseteq [n] \backslash \{i, j\} \big\}.$$

of all elementary CI statements. In this manner, each wall of the S_n-fan is labeled by a CI statement. The map from walls to CI statements is not injective; there are $(n-k-1)!(k-1)!$ walls which are labeled by (6.2).

The S_n-fan is the normal fan [92] of the *permutohedron* $\mathbf{P}_n$, which is the $(n-1)$-dimensional convex hull of the vectors $(\rho_1, \ldots, \rho_n) \in \mathbb{R}^n$, where ρ runs over all rank vectors of permutations in S_n. Each edge of $\mathbf{P}_n$ joins two permutations if they differ by an adjacent transposition. In other words, each edge corresponds to a wall and is thus labeled by a CI statement. A collection of parallel edges of $\mathbf{P}_n$ that are perpendicular to a given hyperplane $\{x_i = x_j\}$ corresponds to the set of CI statements $i \perp\!\!\!\perp j | K$, where K ranges over all subsets of $[n] \backslash \{i, j\}$.

The two-dimensional faces of $\mathbf{P}_n$ are squares and regular hexagons, and two edges of $\mathbf{P}_n$ have the same label in $\mathcal{T}_n$ if, but not only if, they are opposite edges of a square. Figure 2(c) depicts the subset of $\mathbf{P}_5$ in which the last two coordinates of $u \in \mathbb{R}^n$ are less than or equal to all other coordinates. It consists of two copies of the hexagon in 2(a), with the final two entries of the descent vector either 4|5 (in the top hexagon) or 5|4 (in the bottom hexagon). All vertical edges are labeled by the CI statement $4 \perp\!\!\!\perp 5 | \{1, 2, 3\}$.

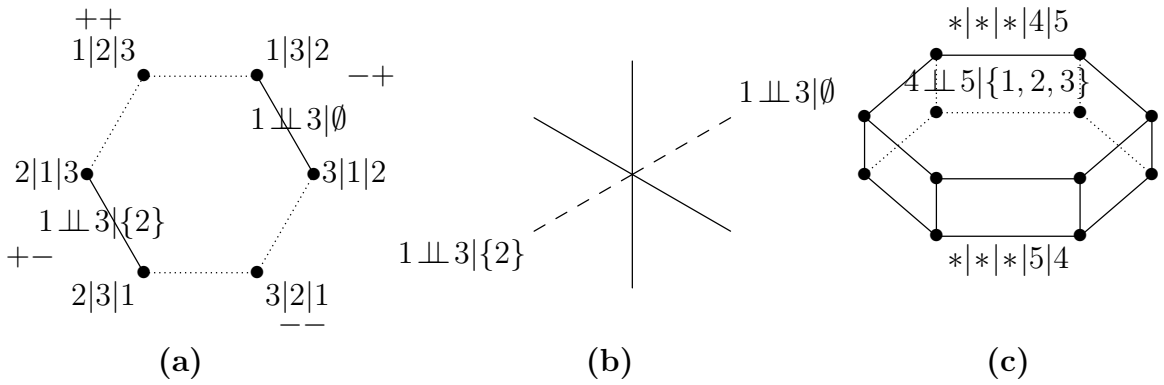


Figure 6.2.: **(a)** The permutohedron $\mathbf{P}_3$ and **(b)** the S_3-fan projected to the plane. The indicated rank test is up-down analysis. Each permutation is represented by its descent vector $\delta = \delta_1 | \delta_2 | \delta_3$. Missing walls of the S_n-fan, or solid edges of $\mathbf{P}_n$, are labeled by CI statements. **(c)** Edges of the permutohedron on opposite sides of a square (here, all vertical edges) are labeled by the same CI statement; hexagonal prisms such as the one pictured here appear in $\mathbf{P}_n$ for $n \geq 5$.

Any convex rank test $\mathcal{F}$ is characterized by the collection of walls $\{\delta, \delta'\}$ that are removed when passing from the S_n-fan to $\mathcal{F}$. So, from (6.2), any convex rank test $\mathcal{F}$ maps to a set $\mathcal{M}_\mathcal{F}$ of CI statements corresponding to missing walls, or a set $\mathbf{M}_\mathcal{F}$ of edges of the permutohedron. For example, if $\mathcal{F}$ is the fan obtained by removing the two dashed rays in Figure 2 (b) then the corresponding set of CI statements is $\mathcal{M}_\mathcal{F} = \{1 \perp\!\!\!\perp 3 | \emptyset,\ 1 \perp\!\!\!\perp 3 | \{2\}\}$.

Conditional independence statements [21] describe the dependence relationship among random variables. A **semigraphoid** is a set $\mathcal{M}$ of *general* conditional independence statements satisfying certain properties [63]. These general conditional independence statements, in contrast to the elementary CI statements already introduced, can take

subsets of $[n]$ in their first two arguments. The conditions are, for X, Y, Z pairwise disjoint subsets of $[n]$,

$$\begin{aligned}
&\text{(SG1)} && X \perp\!\!\!\perp Y \,|\, Z \in \mathcal{M} \implies Y \perp\!\!\!\perp X \,|\, Z \in \mathcal{M} \\
&\text{(SG2)} && X \perp\!\!\!\perp Y \,|\, Z \in \mathcal{M} \text{ and } U \subset X \implies U \perp\!\!\!\perp Y \,|\, Z \in \mathcal{M} \\
&\text{(SG3)} && X \perp\!\!\!\perp Y \,|\, Z \in \mathcal{M} \text{ and } U \subset X \implies X \perp\!\!\!\perp Y \,|\, (U \cup Z) \in \mathcal{M} \\
&\text{(SG4)} && X \perp\!\!\!\perp Y \,|\, Z \in \mathcal{M} \text{ and } X \perp\!\!\!\perp W \,|\, (Y \cup Z) \implies X \perp\!\!\!\perp (W \cup Y) \,|\, Z \in \mathcal{M}.
\end{aligned}$$

It was shown by Studený [79] that these are not a complete set of axioms for probabilistic conditional independence, although they are true of any probabilistic model. A semigraphoid is determined by its *trace* among statements of the form $i \perp\!\!\!\perp j \,|\, K$ where i and j are singletons. Namely, $I \perp\!\!\!\perp J|K$ holds if and only if $i \perp\!\!\!\perp j|L$ for all $i \in I, j \in J$ and L such that $K \subseteq L \subseteq (I \cup J \cup K) \setminus ij$; see [48]. Casting the semigraphoid axiom in terms of the trace, we say that a subset $\mathcal{M}$ of $\mathcal{T}_n$ is a *semigraphoid* if $i \perp\!\!\!\perp j \,|\, K \in \mathcal{M}$ implies $j \perp\!\!\!\perp i \,|\, K \in \mathcal{M}$ and the following axiom holds:

$$\begin{aligned}
\text{(SG)} \qquad & i \perp\!\!\!\perp j \,|\, K \cup \ell \in \mathcal{M} \quad \text{and} \quad i \perp\!\!\!\perp \ell \,|\, K \in \mathcal{M} \\
\text{implies} \quad & i \perp\!\!\!\perp j \,|\, K \in \mathcal{M} \quad \text{and} \quad i \perp\!\!\!\perp \ell \,|\, K \cup j \in \mathcal{M}.
\end{aligned}$$

This axiom is stated in [51, 80]. The first result is that semigraphoids and convex rank tests are the same combinatorial object:

Theorem 6.3.3. *The map $\mathcal{F} \mapsto \mathcal{M}_{\mathcal{F}}$ is a bijection between convex rank tests and semigraphoids.*

Before presenting the proof of this theorem, we shall discuss an example.

Example 6.3.4 (Up-down analysis)**.** *Let $\mathcal{F}$ denote the convex rank test called up-down analysis [90]. In this test, each permutation $\pi \in S_n$ is mapped to the sign vector of its first differences, or, equivalently, its descent set. Thus this test is the natural map $\tau : S_n \to \{-,+\}^{n-1}$. The corresponding semigraphoid $\mathcal{M}_{\mathcal{F}}$ consists of all CI statements $i \perp\!\!\!\perp j \,|\, K$ where $|i-j| \geq 2$.*

This convex rank test is visualized in Figure 2(a,b) for $n = 3$. Permutations are in the same class (have the same sign pattern) if they are connected by a solid edge; there are four classes. In the S_3-fan, the two missing walls are labeled by conditional independence statements as defined in (6.2). For $n = 4$ the up-down analysis test $\mathcal{F}$ is depicted in Figure 3. The double edges correspond to the twelve CI statements in $\mathcal{M}_{\mathcal{F}}$. There are eight classes; e.g., the class $\{3|4|1|2, 3|1|4|2, 1|3|4|2, 1|3|2|4, 3|1|2|4\}$ consists of the five permutations in S_4 which have the up-down pattern $(-,+,-)$.

The proof of Theorem 6.3.3 rests on translating the semigraphoid axiom (SG) into geometric statements about edges of the permutohedron. Recall that a semigraphoid $\mathcal{M}$ can be identified with the set $\mathbf{M}$ of edges of the permutohedron whose CI statement labels are those of $\mathcal{M}$.

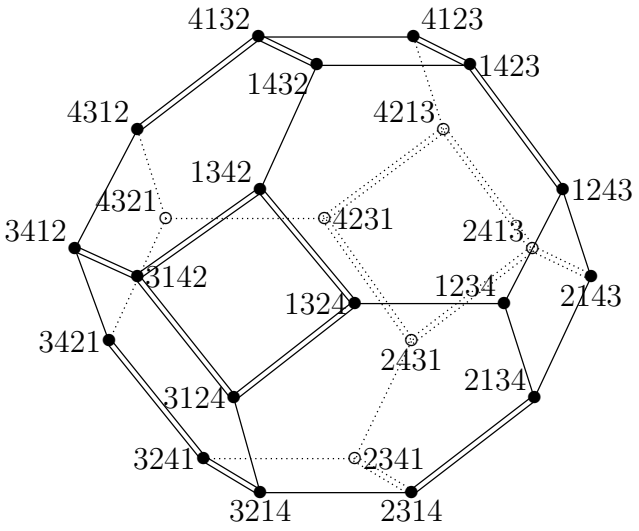
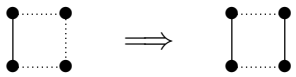
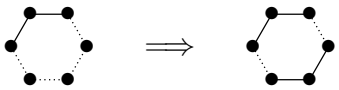


Figure 6.3.: The permutohedron $\mathbf{P}_4$ with vertices marked by descent vectors δ (bars | omitted). The convex rank test indicated by the double edges is up-down analysis.

Observation 6.3.5. *A set* $\mathbf{M}$ *of edges of the permutohedron* $\mathbf{P}_n$ *is a semigraphoid if and only if the set* $\mathbf{M}$ *satisfies the following two geometric axioms:*
Square axiom: *Whenever an edge of a square is in* $\mathbf{M}$*, then the opposite edge is also in* $\mathbf{M}$*.*



Hexagon axiom: *Whenever two adjacent edges of a hexagon are in* $\mathbf{M}$*, then the two opposite edges of that hexagon are also in* $\mathbf{M}$*.*



Let $\mathbf{M}$ be the subgraph of the edge graph of $\mathbf{P}_n$ defined by the statements in $\mathcal{M}$; that is, $\mathbf{M}$ consists of edges whose labels are in $\mathcal{M}$. Each class of the rank test defined by $\mathcal{M}$ consists of the permutations in some connected component of $\mathbf{M}$. We regard a path from δ to δ' on $\mathbf{P}_n$ as a word $\sigma^{(1)} \cdots \sigma^{(l)}$ in the free associative algebra $\mathcal{A}$ generated by the adjacent transpositions of $[n]$. For example, the transposition $\sigma_{23} := (23)$ gives the path from δ to $\delta' = \sigma_{23}\delta = \delta_1|\delta_3|\delta_2|\delta_4|\ldots|\delta_n$. The following relations in $\mathcal{A}$ define a presentation of the group algebra of S_n as a quotient of $\mathcal{A}$:

$$\begin{array}{ll} \text{(BS)} & \sigma_{i,i+1} \cdot \sigma_{i+k+1,i+k+2} - \sigma_{i+k+1,i+k+2} \cdot \sigma_{i,i+1}, \\ \text{(BH)} & \sigma_{i,i+1} \cdot \sigma_{i+1,i+2} \cdot \sigma_{i,i+1} - \sigma_{i+1,i+2} \cdot \sigma_{i,i+1}\sigma_{i+1,i+2}, \qquad \text{and} \\ \text{(BN)} & \sigma_{i,i+1}^2 - 1, \end{array}$$

where suitable i and k vary over $[n]$. The first two are the *braid relations*, and the third represents the idempotency of each transposition.

Now, we regard these relations as properties of a set of edges of $\mathbf{P}_n$, by identifying a word and a permutation δ with the set of edges that comprise the corresponding path in $\mathbf{P}_n$. For example, a set satisfying (BS) is one such that, starting from any δ, the edges of the path $\sigma_{i,i+1}\sigma_{i+k+1,i+k+2}$ are in the set if and only if the edges of the path $\sigma_{i+k+1,i+k+2}\sigma_{i,i+1}$ are in the set. Note then, that (BS) is the square axiom, and (BH) is a weaker version of the hexagon axiom of semigraphoids. That is, implications in either direction hold in a semigraphoid. However, (BN) holds only directionally in a semigraphoid: if an edge lies in the semigraphoid, then its two vertices are in the same class; but the empty path at some vertex δ certainly does not imply the presence of all incident edges in the semigraphoid. Thus, for a semigraphoid, (BS) and (BH) hold, but (BN) must be replaced with the directional version

$$\text{(BN}') \qquad \sigma_{i,i+1}^2 \to 1.$$

We now consider a path p from δ to δ' in a semigraphoid. Here is a crucial lemma for the proof:

Lemma 6.3.6. *Suppose that $\mathcal{M}$ is a semigraphoid. If δ and δ' lie in the same class of $\mathcal{M}$, then so do all shortest paths on $\mathbf{P}_n$ between them.*

The lemma in turn depends on the following version of a classical result due to Jacques Tits. This result, which can be found in [12, p. 49-51]), essentially states that the relations (BS),(BH),(BN) form a Gröbner basis for the two-sided ideals they generate in $\mathcal{A}$.

Theorem 6.3.7 (Tits [84])**.** *Let p and q be words representing paths on $\mathbf{P}_n$.*

(1) A word p is (BS),(BH),(BN)-reduced if and only if it is (BS),(BH),(BN')-reduced.

(2) If p and q are reduced, then they represent the same element of the symmetric group S_n if and only if p can be transformed to q by the application of (BS) and (BH) only.

Proof of Lemma 6.3.6. Theorem 6.3.7 (1) says that if there is any path connecting δ and δ', then there is a shortest path connecting them. Thus if δ and δ' lie in the same class of $\mathcal{M}$, some shortest path $\delta \to \delta'$ also lies in that class. Now (2) says that if p and q are both shortest paths, then q can be obtained from p by application of only the square and hexagon axioms, (BS) and (BH). Thus if any shortest path $\delta \to \delta'$ lies in the class of $\mathcal{M}$ containing them both, so do all other shortest paths connecting them. □

We need one lemma to deal with intersections of nonmaximal cones. Denote by $\prec$ the transitive relation "is a face of" and write $F_w(C)$ for the face of a cone C at which w is minimized.

Lemma 6.3.8. *If the intersection of two cones C_1 and C_2 is a face of both, then the intersection of any faces $D \prec C_1$ and $E \prec C_2$ is a face of both.*

Proof. By transitivity of $\prec$ and the hypothesis it suffices to show $D \cap E \prec C_1 \cap C_2$. Since $D \prec C_1$, there exists a linear functional w such that the face $F_w(C_1)$ equals D and $C_1 \cap C_2 \subset C_1 \subset H_w^+$. Then $F_w(C_1 \cap C_2) = D \cap C_2$ so $D \cap C_2 \prec C_1 \cap C_2$. Similarly, $E \cap C_1 \prec C_1 \cap C_2$. Then since the intersection of any two faces of $C_1 \cap C_2$ is also a face, $D \cap E \prec C_1 \cap C_2$ as desired. □

Proof of Theorem 6.3.3. Both semigraphoids and convex rank tests can be regarded as sets of edges of $\mathbf{P}_n$. We first show that a semigraphoid satisfies (PC). Consider δ, δ' in the same class C of a semigraphoid, and let $\delta'' \in \mathcal{L}(\delta \cap \delta')$. Further, let p be a shortest path from δ to δ'' (so, $p\delta = \delta''$), and let q be a shortest path from δ'' to δ'. We claim that qp is a shortest path from δ to δ', and thus $\delta'' \in C$ by Lemma 6.3.6. Suppose qp is not a shortest path. Then, we can obtain a shorter path in the semigraphoid by some sequence of substitutions according to (BS), (BH), and (BN'). Only (BN') decreases the length of a path, so the sequence must involve (BN'). Therefore, there is some i, j in $[n]$, such that their positions relative to each other are reversed twice in qp. But p and q are shortest paths, hence one reversal occurs in each of p and q. Then δ and δ' agree on whether $i > j$ or $j > i$, but the reverse holds in δ'', contradicting $\delta'' \in \mathcal{L}(\delta \cap \delta')$. Thus every semigraphoid is a pre-convex rank test.

Now, we show that a semigraphoid corresponds to a fan. We first argue that we may reduce to the case of two maximal cones, each coming from a class in the semigraphoid, whose intersection is codimension one in both. By Lemma 6.3.8, we can consider maximal cones only. Suppose two maximal cones C_1, C_k have intersection $C_1 \cap C_k$ which is not codimension one. Then there exists a sequence of maximal cones $C_1, C_2, \ldots, C_k$ such that $C_i \cap C_{i+1}$ is codimension one, $C_1 \cap C_k \subset C_i \cap C_{i+1}$ for all $i = 1, \ldots k-1$, and in fact $C_1 \cap C_k = C_1 \cap C_2 \cap \cdots \cap C_k$. We have that $(C_i \cap C_{i+1}) \cap (C_{i+1} \cap C_{i+2})$ is a face of C_{i+1} and C_{i+2} by Lemma 6.3.8, and also is a face of C_i. Thus $C_i \cap C_{i+1} \cap C_{i+2} \prec C_i, C_{i+1}, C_{i+2}$; continuing in this manner, we eventually get that $C_1 \cap C_2 \cap \cdots \cap C_k \prec C_1, C_k$ as required.

Consider the cone corresponding to a class C. We need only show that its codimension one intersection with another maximal cone is a shared face. Since C is a cone of a coarsening of the S_n-fan, each facet of C lies in a hyperplane $H = \{x_i = x_j\}$. Suppose a face of C coincides with the hyperplane H and that $i > j$ in C. A vertex δ borders H if i and j are adjacent in δ. We will show that if $\delta, \delta' \in C$ border H, then their reflections $\widehat{\delta} = \delta_1 | \ldots | j | i | \ldots | \delta_n$ and $\widehat{\delta'} = \delta'_1 | \ldots | j | i | \ldots | \delta'_n$ both lie in some class C'. Consider a 'great circle' path between δ and δ' which stays closest to H: all vertices in the path have i and j separated by at most one position, and no two consecutive vertices have i and j nonadjacent. This is a shortest path, so it lies in C, by Lemma 6.3.6. Using the square and hexagon axioms (Observation 6.3.5), we see that the reflection of the path across H is a path in the semigraphoid that connects $\widehat{\delta}$ to $\widehat{\delta'}$ (Figure 3). This shows that the intersection of C and C' is a face of both. Thus a semigraphoid is a convex rank test.

Finally, if $\mathbf{M}$ is a set of edges of $\mathbf{P}_n$ representing a convex rank test, then it is easy to show that $\mathbf{M}$ satisfies the square and hexagon axioms. □

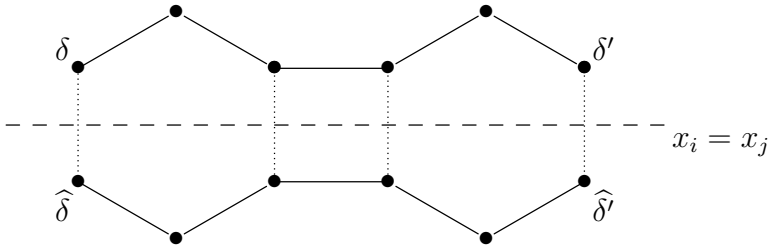


Figure 6.4.: Reflecting a path across a hyperplane.

6.4. The submodular cone

In this section we focus on a subclass of the convex rank tests. Let $2^{[n]}$ denote the collection of all subsets of $[n] = \{1, 2, \ldots, n\}$. Any real-valued function $w : 2^{[n]} \to \mathbb{R}$ defines a convex polytope Q_w of dimension $\leq n-1$ as follows:

$$\begin{aligned} Q_w \;:=\; & \big\{ x \in \mathbb{R}^n \,:\, x_1 + x_2 + \cdots + x_n = w([n]) \\ & \text{and } \textstyle\sum_{i \in I} x_i \leq w(I) \text{ for all } \emptyset \neq I \subseteq [n] \big\}. \end{aligned}$$

A function $w : 2^{[n]} \to \mathbb{R}$ is called **submodular** if $w(I) + w(J) \geq w(I \cap J) + w(I \cup J)$ for $I, J \subseteq [n]$. The **submodular cone** is the cone $\mathbf{C}_n$ of all submodular functions $w : 2^{[n]} \to \mathbb{R}$. Working modulo its lineality space $\mathbf{C}_n \cap (-\mathbf{C}_n)$, we regard $\mathbf{C}_n$ as a pointed cone of dimension $2^n - n - 1$.

Studying functions w means that in considering the normal fan of a polytope Q_w, we want to retain information about non-binding inequalities that are just barely so, i.e. that hold with equality. For this reason we define the *vector (normal) fan* [5]. The indicator function of each $I \in 2^{[n]}$ defines a vector e_I in the 1-skeleton of the S_n-fan, understood modulo $e_{[n]}$; for example, these vectors for $n = 3$ are $e_{001}, e_{010}, e_{100}, e_{011}, \ldots, e_{111}$.

A **vector fan** $\mathfrak{F}$ is a collection of subsets of $\{e_I : I \in 2^{[n]}\}$ such that $U, V \in \mathfrak{F}$ implies $U \cap V \in \mathfrak{F}$. A vector fan defines a usual fan by taking the maximal cones of the fan to be the cones generated by the vector sets in the vector fan. We say that a vector fan is *complete* if its fan is. A vector fan $\mathfrak{F}$ *coarsens* another vector fan $\mathfrak{G}$ if for all $U \in \mathfrak{G}$, there exists $V \in \mathfrak{F}$ with $U \subset V$.

Given a function $w : 2^{[n]} \to \mathbb{R}$, each $I \in 2^{[n]}$ defines an inequality $\sum_{i \in I} x_i \leq w_I$ appearing in the definition of Q_w; the vector normal fan tells us which of these inequalities holds with equality on some face of Q_w. We define the **vector normal fan** of a function $w : 2^{[n]} \to \mathbb{R}$ as the set $\{\{e_I : I \in 2^{[n]}, \sum_{i \in I} x_i = w_I \text{ for all } x \in F\}$ for each face $F \in Q_w\}$. The vector normal fan of w defines a fan which is the normal fan of Q_w and retains additional information.

Proposition 6.4.1. *A function $w : 2^{[n]} \to \mathbb{R}$ is submodular if and only if the vector normal fan of w is a coarsening of the vector S_n-fan.*

Example 6.4.2. *Let $w_1 = w_2 = w_3 = 1, w_{12} = w_{13} = w_{23} = w_{123} = 3$. The polytope Q_w is the point $(1, 1, 1)$ but the function w is not submodular. The vector normal fan $\mathfrak{F}$ of*

w is $\{\{e_{001}, e_{010}, e_{100}\}\}$ and the normal fan is all of $\mathbb{R}^3/(1,1,1)$. $\mathfrak{F}$ does not coarsen the S_n-fan since, for example, e_{110} is not contained in any set in $\mathfrak{F}$.

However, if we change w slightly to define the same Q_w but with the inequalities corresponding to $011, 101$, and 110 also holding with equality, e.g. $w_1 = w_2 = w_3 = 1, w_{12} = w_{13} = w_{23} = 2$, and $w_{123} = 3$, the resulting vector normal fan of w is a coarsening of the (vector) S_n-fan.

Proof. We show only the if direction of Proposition 6.4.1. Suppose w is not submodular. Then there exist $I, J \subset 2^{[n]}$ such that

$$w_I + w_J < w_{I \cap J} + w_{I \cup J}$$

We also have that

$$\begin{aligned} \sum_{i \in I \cup J} x_i + \sum_{i \in I \cap J} x_i &= \sum_{i \in I} x_i + \sum_{i \in J} x_i \\ &\leq w_I + w_J < w_{I \cap J} + w_{I \cup J} \end{aligned}$$

So $\sum_{i \in I \cup J} x_i < w_{I \cup J} + (w_{I \cap J} - \sum_{i \in I \cap J} x_i)$ and similarly $\sum_{i \in I \cap J} x_i < w_{I \cap J} + (w_{I \cup J} - \sum_{i \in I \cup J} x_i)$, so that at most one of the inequalities corresponding to $I \cup J$ and $I \cap J$ can hold with equality at any point of Q_w. Then any set in the vector normal fan of w either fails to contain $e_{I \cap J}$ or fails to contain $e_{I \cup J}$. □

Proposition 6.4.1 can be paraphrased as follows: the function w is submodular if and only if the optimal solution of

$$\text{maximize } u \cdot x \text{ subject to } x \in Q_w$$

depends only on the permutation equivalence class of u. Thus, solving this linear programming problem constitutes a convex rank test. Any such test is called a *submodular rank test.*

A convex polytope is a *(Minkowski) summand* of another polytope if the normal fan of the latter refines the normal fan of the former. The polytope Q_w that represents a submodular rank test is a summand of the permutohedron $\mathbf{P}_n$.

Theorem 6.4.3. *The following combinatorial objects are equivalent for any positive integer n:*
1. *submodular rank tests,*
2. *summands of the permutohedron $\mathbf{P}_n$,*
3. *structural conditional independence models [80],*
4. *faces of the submodular cone $\mathbf{C}_n$ in $\mathbb{R}^{2^n}$.*

Proof. We have $1 \iff 2$ from Proposition 6.4.1, and $1 \iff 3$ follows from [80]. Further, $1 \iff 4$ is a direct consequence of the definition of submodular rank tests. □

Remark 6.4.4. *All 22 convex rank tests for $n = 3$ are submodular. The submodular cone $\mathbf{C}_3$ is a 4-dimensional cone whose base is a bipyramid. Its f-vector is $(1, 5, 9, 6, 1)$. The polytopes Q_w, as w ranges over representatives of the faces of $\mathbf{C}_3$, are all the Minkowski summands of $\mathbf{P}_3$.*

Proposition 6.4.5. *For $n \geq 4$, there exist convex rank tests that are not submodular rank tests. Equivalently, there are fans that coarsen the S_n-fan but are not the normal fan of any polytope.*

Proof. This result is well-known. It is stated in Section 2.2.4 of [80] in the following form: "There exist semigraphoids that are not structural." □

An interesting example which also proves Proposition 6.4.5 is the following semigraphoid:

$$\mathcal{M} \quad = \quad \big\{2 \perp\!\!\!\perp 3|\{1,4\},\, 1 \perp\!\!\!\perp 4|\{2,3\},\, 1 \perp\!\!\!\perp 2|\emptyset,\, 3 \perp\!\!\!\perp 4|\emptyset\,\big\}.$$

The corresponding fan consists of unimodular cones, or, equivalently, the posets P_i representing this non-submodular convex rank test are all trees. This example answers a question posed in the first version of [66]. A systematic method for showing that a semigraphoid is not submodular is described in Chapter 7. Results include an example of a coarsest semigraphoid which is not submodular and a proof that the semigraphoid semigroup is not normal.

Remark 6.4.6. *For $n = 4$ there are* 22108 *submodular rank tests, one for each face of the* 11*-dimensional cone $\mathbf{C}_4$. The base of this submodular cone is a* 10*-dimensional polytope with f-vector* $(1, 37, 356,\ 1596, 3985, 5980, 5560, 3212, 1128, 228, 24, 1)$. *The* 37 *vertices of this polytope correspond to the maximal semigraphoids. These come in seven symmetry classes up to the $*$ involution (6.3) and the S_4-action. The types of maximal semigraphoids for $n = 4$ are displayed in the following table:*

Symmetry	*No.*	$i \perp\!\!\!\perp j$	$i \perp\!\!\!\perp j \mid k$	$i \perp\!\!\!\perp j \mid \{k,l\}$
$1\times$ *and* $*$	*2*	*all*	*all*	*none*
$4\times$ *and* $*$	*8*	*all*	*all but* $2 \perp\!\!\!\perp 3|1, 1 \perp\!\!\!\perp 3|2, 1 \perp\!\!\!\perp 2|3$	$3 \perp\!\!\!\perp 4|12, 2 \perp\!\!\!\perp 4|13, 1 \perp\!\!\!\perp 4|23$
$6\times$ *incl.* $*$	*6*	*all but* $1 \perp\!\!\!\perp 2$	*all but* $1 \perp\!\!\!\perp 2|3, 1 \perp\!\!\!\perp 2|4$	*all but* $1 \perp\!\!\!\perp 2|34$
$4\times$ *and* $*$	*8*	*all*	$2 \perp\!\!\!\perp 3|4, 2 \perp\!\!\!\perp 4|3, 3 \perp\!\!\!\perp 4|2$	$3 \perp\!\!\!\perp 4|12, 2 \perp\!\!\!\perp 4|13, 2 \perp\!\!\!\perp 3|14$
$1\times$*, self-*$*$*	*1*	*all*	*none*	*all*
$6\times$ *incl.* $*$	*6*	*all but* $1 \perp\!\!\!\perp 2$	$2 \perp\!\!\!\perp 3|1, 2 \perp\!\!\!\perp 4|1, 1 \perp\!\!\!\perp 3|2, 1 \perp\!\!\!\perp 4|2$	*all but* $3 \perp\!\!\!\perp 4|12$
$6\times$ *incl.* $*$	*6*	$3 \perp\!\!\!\perp 4$	*all but* $2 \perp\!\!\!\perp 3|4, 2 \perp\!\!\!\perp 4|3, 1 \perp\!\!\!\perp 4|3, 1 \perp\!\!\!\perp 3|4$	$1 \perp\!\!\!\perp 2|34$

Remark 6.4.7. *For $n = 5$ there are* 117978 *coarsest submodular rank tests, in* 1319 *S_5 symmetry classes. We confirmed this result of [81] with* `polymake` *[30].*

We now define a class of submodular rank tests, which we call *Minkowski sum of simplices (MSS) tests*. Note that each subset K of $[n]$ defines a submodular function w_K by setting $w_K(I) = 1$ if $K \cap I$ is non-empty and $w_K(I) = 0$ if $K \cap I$ is empty. The corresponding polytope Q_{w_K} is the simplex $\Delta_K = \mathrm{conv}\{e_k : k \in K\}$.

Now consider an arbitrary subset $\mathcal{K} = \{K_1, K_2, \ldots, K_r\}$ of $2^{[n]}$. It defines the submodular function $w_{\mathcal{K}} = w_{K_1} + w_{K_2} + \cdots + w_{K_r}$. The corresponding polytope is the Minkowski sum

$$\Delta_{\mathcal{K}} \quad = \quad \Delta_{K_1} + \Delta_{K_2} + \cdots + \Delta_{K_r}.$$

The associated MSS test $\tau_{\mathcal{K}}$ is defined as follows. Given $\rho \in S_n$, we compute the number of indices $j \in [r]$ such that $\max\{\rho_k : k \in K_j\} = \rho_i$, for each $i \in [n]$. The signature $\tau_{\mathcal{K}}(\rho)$ is the vector in $\mathbb{N}^n$ whose ith coordinate is that number. Few submodular rank tests are MSS tests:

Remark 6.4.8. *For $n = 3$, there are 22 submodular rank tests, but only 15 of them are MSS tests. For $n = 4$, there are 22108 submodular rank tests, but only 1218 of them are MSS tests.*

In light of Theorem 6.3.3, it is natural to ask which semigraphoids correspond to an MSS test. Geometrically, we wish to know which edges of the permutohedron $\mathbf{P}_n$ are contracted when passing to the polytope $Q_{w_{\mathcal{K}}}$. To be precise, let $\mathcal{M}_{\mathcal{K}}$ denote the semigraphoid derived from $\mathcal{F}_{w_{\mathcal{K}}}$ using the bijection in Theorem 6.3.3. We then have the following result:

Proposition 6.4.9. *The semigraphoid $\mathcal{M}_{\mathcal{K}}$ is the set of CI statements of the form $i \perp\!\!\!\perp j \,|\, K$ where all sets containing $\{i,j\}$ and contained in $\{i,j\} \cup [n]\backslash K$ are not in $\mathcal{K}$.*

Proof. Consider two permutations δ and δ' which are adjacent on the permutohedron $\mathbf{P}_n$, and let $i \perp\!\!\!\perp j \,|\, K$ be the label of the edge that connects δ and δ'. That CI statement is in $\mathcal{M}_{\mathcal{K}}$ if and only if δ and δ' are mapped to the same vertex in $\Delta_{\mathcal{K}}$ if and only if δ and δ' are mapped to the same vertex in each simplex Δ_{K_l} for $l = 1, 2, \ldots, r$. For each l, this means that the leftmost entry of the descent vector δ that lies in K_l agrees with the leftmost entry of the other descent vector δ' that lies in K_l. This condition is equivalent to

$$K_l \,\cap\, (\, K \,\cup\, \{i,j\}\,) \quad \neq \quad \{i,j\} \qquad\qquad \text{for } \; l = 1, 2, \ldots, r.$$

Thus $i \perp\!\!\!\perp j \,|\, K$ is in the semigraphoid $\mathcal{M}_{\mathcal{K}}$ associated with the set family $\mathcal{K}$ if and only if $\mathcal{K}$ contains no set whose intersection with $K \cup \{i,j\}$ equals $\{i,j\}$. This is precisely the claim. □

There is a natural involution $*$ on the set of all CI statements which is defined as follows:

$$(i \perp\!\!\!\perp j \,|\, C)^* \quad := \quad i \perp\!\!\!\perp j \,|\, [n]\backslash(C \cup \{i,j\}). \tag{6.3}$$

If $\mathcal{M}$ is any semigraphoid, then the semigraphoid $\mathcal{M}^*$ is obtained by applying the involution $*$ to all the CI statements in the model $\mathcal{M}$. This involution is referred to as *duality* in [47]. In the *boolean lattice*, whose elements are the subsets of $[n]$, the involution corresponds to switching the role of set intersection and set union.

The MSS test $\tau_{\mathcal{K}}$ was defined above in terms of weight functions w. What follows is a similar construction for the duals of MSS tests. Let $z_{\mathcal{K}}(J) = 1$ for $J \in \mathcal{K}$ and $z_{\mathcal{K}}(J) = 0$ otherwise. Then the function $w^* : 2^{[n]} \to \mathbb{R}$ defined by $w^*_{\mathcal{K}}(I) := \sum_{J \subset I} z_{\mathcal{K}}(J)$ is supermodular. We set

$$\begin{aligned} Q^*_w \;:= \big\{\, & x \in \mathbb{R}^n \,:\, x_1 + x_2 + \cdots + x_n = w([n]) \\ & \text{and } \textstyle\sum_{i \in I} x_i \geq w(I) \text{ for all } \emptyset \neq I \subseteq [n] \,\big\}. \end{aligned}$$

Then the equality $Q^*_{w_{\mathcal{K}}} = \Delta_K$ holds for $\Delta_{\mathcal{K}} = \Delta_{\mathcal{K}_1} + \Delta_{K_2} + \cdots + \Delta_{K_r}$. This equality is precisely the statement in Proposition 6.3 of Postnikov's paper [65].

6.5. Graphical tests

We have seen that semigraphoids are equivalent to convex rank tests. We now explore the connection to graphical models. Let G be a graph with vertex set $[n]$ and $\mathcal{K}(G)$ the collection of all subsets $K \subseteq [n]$ such that the induced subgraph of $G|_K$ is connected. The *undirected graphical model* (or *Markov random field*) derived from the graph G is the set $\mathcal{M}^G$ of CI statements:

$$\mathcal{M}^G \;=\; \big\{\, i \perp\!\!\!\perp j \,|\, C \;:\; \text{the restriction of } G \text{ to } [n]\backslash C \text{ contains no path from } i \text{ to } j \big\}. \tag{6.4}$$

Theorem 6.5.1. *The set $\mathcal{M}^G$ of CI statements in the graphical model G is equal to the semigraphoid $\mathcal{M}_{\mathcal{K}(G)}$ associated with the family $\mathcal{K}(G)$ of connected induced subgraphs of G.*

Proof. The defining condition in (6.4) is equivalent to saying that the restriction of G to any node set containing $\{i, j\}$ and contained in $\{i, j\} \cup ([n]\backslash C)$ is disconnected. With this observation, Theorem 6.5.1 follows directly from Proposition 6.4.9. □

The polytope $\Delta_G = \Delta_{\mathcal{K}(G)}$ associated with the graph G is the *graph associahedron*. This is a well-studied object in combinatorics [65, 18]. Carr and Devadoss [18] showed that Δ_G is a simple polytope whose faces are in bijection with the tubings of the graph G. Tubings are defined as follows. Two subsets $A, B \subset [n]$ are *compatible* for G if one of the following conditions holds: $A \subset B$, $B \subset A$, or $A \cap B = \emptyset$, and there is no edge between any node in A and B. A *tubing* of the graph G is a subset $\mathbf{T}$ of $2^{[n]}$ such that any two elements of $\mathbf{T}$ are compatible. The set of all tubings on G is a simplicial complex; it is dual to the face lattice of the simple polytope Δ_G.

For any graph G on $[n]$ we now have two convex rank tests. First, there is the *graphical model rank test* $\tau_{\mathcal{K}(G)}$, which is the MSS test of the set family $\mathcal{K}(G)$. Second, we have the *graphical tubing rank test* $\tau^*_{\mathcal{K}(G)}$, which is the convex rank test associated with the semigraphoid $(\mathcal{M}^G)^*$ dual to $\mathcal{M}^G$. Explicitly, that dual semigraphoid is given by

$$(\mathcal{M}^G)^* \;= \\ \big\{\, i \perp\!\!\!\perp j \,|\, C \;:\; \text{the restriction of } G \text{ to } C \cup \{i, j\} \text{ contains no path from } i \text{ to } j \big\}. \tag{6.5}$$

We summarize the discussion in the following theorem:

Theorem 6.5.2. *The following four combinatorial objects are isomorphic for any graph G on $[n]$:*
- *the graphical model rank test $\tau_{\mathcal{K}(G)}$,*
- *the graphical tubing rank test $\tau^*_{\mathcal{K}(G)}$,*
- *the fan of the graph associahedron Δ_G,*
- *the simplicial complex of all tubings on G.*

We note that when the graph G is a path of length n, Δ_G is the *associahedron*, and when it is an n-cycle, Δ_G is the *cyclohedron*. The number of classes in either the MSS

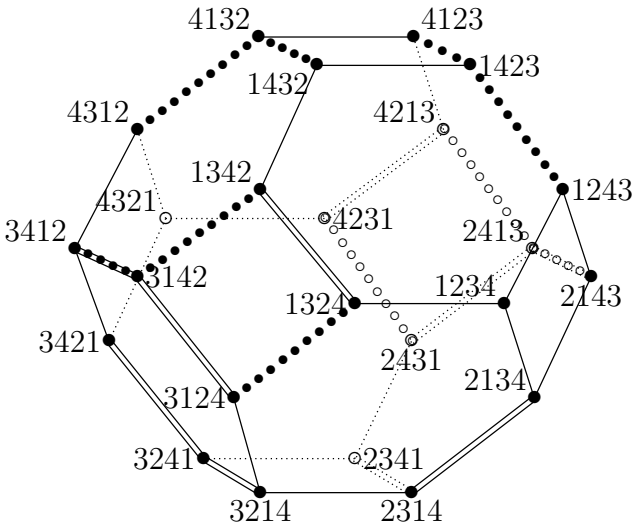


Figure 6.5.: The permutohedron $\mathbf{P}_4$. Double edges indicate the MSS test $\tau_{\mathcal{K}(G)}$ where G is the 4-chain. Edges with large dots indicate the dual tubing test $\tau^*_{\mathcal{K}(G)}$.

test $\tau_{\mathcal{K}(G)}$ or the tubing test $\tau^*_{\mathcal{K}(G)}$ is the *G-Catalan number* of [65]. This number is the classical Catalan number $\frac{1}{n+1}\binom{2n}{n}$ for the associahedron test. It equals $\binom{2n-2}{n-1}$ for the cyclohedron test [56].

Example 6.5.3. *Let $n = 4$ and let G be the 4-chain 1—2—3—4. Then*

$$\begin{aligned}\mathcal{M}^G &= \{1 \perp\!\!\!\perp 3\,|\,24,\ \ 1 \perp\!\!\!\perp 4\,|\,23,\ \ 2 \perp\!\!\!\perp 4\,|\,13,\ \ 1 \perp\!\!\!\perp 3\,|\,2,\ \ 1 \perp\!\!\!\perp 4\,|\,2,\ \ 1 \perp\!\!\!\perp 4\,|\,3,\ \ 2 \perp\!\!\!\perp 4\,|\,3\},\\ (\mathcal{M}^G)^* &= \{1 \perp\!\!\!\perp 3,\ \ \ 1 \perp\!\!\!\perp 4,\ \ \ 2 \perp\!\!\!\perp 4,\ \ \ 1 \perp\!\!\!\perp 3\,|\,4,\ \ 1 \perp\!\!\!\perp 4\,|\,3,\ \ 1 \perp\!\!\!\perp 4\,|\,2,\ \ 2 \perp\!\!\!\perp 4\,|\,1\}.\end{aligned}$$

The corresponding tests $\tau_{\mathcal{K}(G)}$ and $\tau^*_{\mathcal{K}(G)}$ are depicted in Figure 6.5. Note that contracting either class of marked edges on the permutohedron in Figure 6.5 leads to the 3-dimensional associahedron Δ_G. The associahedron Δ_G is the Minkowski sum of the simplices Δ_K where K runs over

$$\mathcal{K}(G) \quad = \quad \{\{1\},\{2\},\{3\},\{4\},\{1,2\},\{2,3\},\{3,4\},\{1,2,3\},\{2,3,4\},\{1,2,3,4\}\}.$$

The 3-dimensional simple polytope Δ_4 has 14 vertices, one for each of the 14 tubings of G. □

In the application of graphical rank tests, we found it more natural to work with the tubing test $\tau^*_{\mathcal{K}(G)}$ instead of the MSS test $\tau_{\mathcal{K}(G)}$. We refer to [56] which gives a detailed discussion of the cyclohedron test and its applications. By the cyclohedron test we mean the tubing test $\tau^*_{\mathcal{K}(G)}$ where the graph G is a cycle of length n.

Applying the tubing test to a data vector $u \in \mathbb{R}^n$ can be viewed as an iterative procedure for drawing a topographic map on the graph G. Namely, we encircle the vertices of G by sets $U_1, \ldots, U_n$ in the order $\delta_1, \delta_2, \ldots, \delta_{n-1}$, with the following provision: if δ_i is next to be encircled and shares an edge with some vertex j which has already

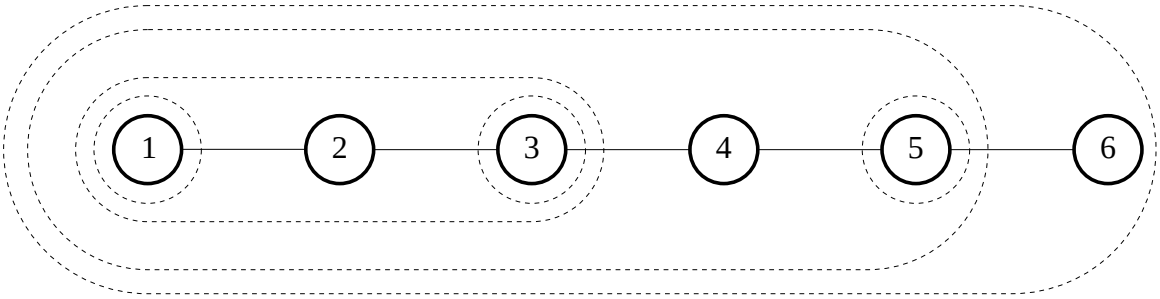


Figure 6.6.: Tubing of the 6-chain. Encircled regions indicate the sets U_j.

been encircled by some U_j, then U_i must also contain the circle U_j. The result is a collection U of $n-1$ encircled sets $U_1, U_2, \ldots, U_{n-1}$, and this unordered collection of sets is the signature of v. The height h_i of the i-th node in the topographic map for v is the number of sets U_j which contain i. We can identify the signature U with the *height vector* $h = (h_1, h_2, \ldots, h_n)$, since U can be recovered uniquely from the vector h. The map $u \mapsto h(u)$ can be interpreted as a *smoothing of the data*. Figure 6 displays the topographic map when the data vector is $u = (2.1, 0.3, 1.8, , 2.0, 1.1, 0.1)$. Here G is the 6-chain 1—2—3—4—5—6. and the descent vector of u equals $\delta = (1|5|3|2|4|6)$.

Conclusions

This work describes the connections among algebraic combinatorics, non-parametric statistics and graphical models (statistical learning theory). Specifically, we have proved the equivalence between semigraphoids and convex rank tests. This result provides the background for the counterexamples given in the next chapter and the rank tests which were applied to biological data in [56].

Chapter 7.

Three counterexamples on semigraphoids

Semigraphoids are combinatorial structures that arise in statistical learning theory. They are equivalent to convex rank tests and to polyhedral fans that coarsen the reflection arrangement of the symmetric group S_n. In this chapter we resolve two problems on semigraphoids posed in Studený's book [80], and we answer a related question by Postnikov, Reiner, and Williams on generalized permutohedra [66]. We also study the semigroup and the toric ideal associated with semigraphoids.

This chapter is published as "Three Counterexamples on Semigraphoids" [40] in the Journal Combinatorics, Probability and Computing together with Raymond Hemmecke, Jason Morton, Anne Shiu and Bernd Sturmfels.

Jason Morton and Bernd Sturmfels were supported by the DARPA *Fundamental Laws of Biology* program, and Bernd Sturmfels was also supported by the NSF. Anne Shiu was supported by a Lucent Technologies Bell Labs Graduate Research Fellowship. Oliver Wienand was supported by the Wipprecht foundation.

7.1. Introduction

A **conditional independence (CI) statement** on a finite set of random variables, indexed by $[n] = \{1, 2, \ldots, n\}$, is a formal symbol $[i \perp\!\!\!\perp j \,|\, K]$ where $K \subset [n]$ and $i, j \in [n] \backslash K$. The symbol $[i \perp\!\!\!\perp j \,|\, K]$ represents the statement that the random variables i and j are conditionally independent given the joint random variable K. For any joint probability distribution on the n random variables, the set $\mathcal{M}$ of all CI statements that are valid for the given distribution satisfies the following axiom:

(**SG**) If $[i \perp\!\!\!\perp j \,|K \cup \ell]$ and $[i \perp\!\!\!\perp \ell \,|K]$ are in $\mathcal{M}$ then so are $[i \perp\!\!\!\perp j \,|K]$ and $[i \perp\!\!\!\perp \ell \,|K \cup j]$.

A **semigraphoid** is any set $\mathcal{M}$ of CI statements which satisfies the axiom (**SG**). Studený's book [80] gives an introduction to semigraphoids and their role in statistical learning theory. For further details and references see also Matúš [49, 52]. In this paper we construct examples which answer two problems stated by Studený:

(**Q1**) *Is it true that every coatom of the lattice of (disjoint) semigraphoids over* $[n]$ *is a structural independence model over* $[n]$*?*

[80, Question 4, page 194]

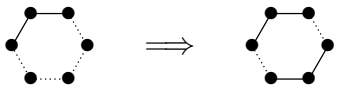
(**Q2**) *Is every structural imset over* $[n]$ *already a combinatorial imset over* $[n]$*?*

[80, Question 7, page 207]

The approach is based on the geometric characterization of semigraphoids which was developed in [54]. Let Π_{n-1} denote the $(n-1)$-dimensional *permutohedron* [50, 92], and let $C_n = [0,1]^n$ denote the standard n-dimensional cube. The vertices of Π_{n-1} are in bijection with the elements of the symmetric group S_n, and with the monotone edge paths from $(0,0,\ldots,0)$ to $(1,1,\ldots,1)$ on the cube C_n. The 2-dimensional faces of C_n are in bijection with the CI statements on $[n]$. Namely, $[i \perp\!\!\!\perp j \,|\, K] = [j \perp\!\!\!\perp i \,|\, K]$ represents the 2-face of C_n with $x_k = 1$ for $k \in K$ and $x_l = 0$ for $l \in [n]\backslash(K \cup \{i,j\})$. The number of these 2-cubes equals $\gamma_n := \binom{n}{2}2^{n-2}$. There is a natural surjection from the edges of Π_{n-1} onto the 2-faces of C_n. Namely, an edge of Π_{n-1} corresponds to a pair of adjacent monotone edge paths on C_n. These adjacent paths differ only along a 2-cube $[i \perp\!\!\!\perp j \,|\, K]$. In this manner, we identify any set $\mathcal{M}$ of CI statements on $[n]$ with a set of 2-cubes on the boundary of C_n. We also identify $\mathcal{M}$ with a set of edges of the permutohedron Π_{n-1}, bearing in mind that opposite edges of a square have the same CI statement as their label.

Each 2-face of the permutohedron Π_{n-1} is either a square or a hexagon. By [54], the semigraphoid axiom is equivalent to the following geometric condition on Π_{n-1}:

(**SG′**) *If two adjacent edges of a hexagon are in* $\mathcal{M}$ *then so are their two opposites.*



The normal fan of the permutohedron Π_{n-1} is the reflection arrangement of S_n. Theorem 3 in [54] identifies semigraphoids with fans that coarsen this arrangement. Such fans are called *convex rank tests*. Namely, $\mathcal{M}$ specifies the set of edges of Π_{n-1} whose dual walls in the normal fan are not present in the convex rank test.

A basic question about any semigraphoid $\mathcal{M}$ is whether its corresponding convex rank test is *submodular*, in other words, whether it is the normal fan of a convex polytope. That polytope would then be a Minkowski summand of Π_{n-1}. These polytopes are known as *generalized permutohedra* and they were studied in [65, 66].

Studený's first question has the following geometric translations:

(**Q1**) *Is every coarsest convex rank test submodular?*

(**Q1**) *Is every fan which maximally coarsens the* S_n*-arrangement the normal fan of a generalized permutohedron?*

In the first version of [66], Postnikov, Reiner and Williams asked a similar question:

(**Q3**) *Is every simplicial fan which coarsens the* S_n*-arrangement the normal fan of a simple generalized permutohedron?*

This paper answers all three questions. In Section 2 we derive and explain the counterexample for Question (**Q3**). That example is discussed in [66, Example 3.8]. By Studený's classification of the 26424 semigraphoids for $n = 4$, it had been known that the answers to Questions (**Q1**) and (**Q2**) are affirmative for $n \leq 4$. In Sections 3 and 4 we construct counterexamples for (**Q1**) and (**Q2**) with $n = 5$.

Question (**Q2**) has the following reformulation in the setting of toric algebra [53, §7]. We represent the semigraphoid axiom as an equation among formal symbols:

$$(\mathbf{SG}'') \qquad [i \perp\!\!\!\perp j \,|K \cup \ell] \,+\, [i \perp\!\!\!\perp \ell \,|K] \quad = \quad [i \perp\!\!\!\perp j \,|K] \,+\, [i \perp\!\!\!\perp \ell \,|K \cup j]$$

for all i, j, l, K. These relations span the kernel of the linear map

$$\mathcal{A} \,:\, \mathbb{Z}^{\gamma_n} \,\rightarrow\, \mathbb{Z}^{2^n}\,,\;\; [i \perp\!\!\!\perp j \,|\, K] \;\;\mapsto\;\; e_{iK} + e_{jK} - e_K - e_{ijK}. \tag{7.1}$$

A semigraphoid is a solution to the equations (**SG**″) in the semiring $\{0, +\}$, representing "zero" and "positive". A semigraphoid is *submodular* if it is the set of zero coordinates of a solution to (**SG**″) in the non-negative real numbers. These definitions furnish us with an algebraic representation of a semigraphoid $\mathcal{M}$ and a systematic method for testing submodularity of $\mathcal{M}$ by linear programming. Studený's question (**Q2**) concerns the $\mathbb{N}$-linear span of the columns of the matrix $\mathcal{A}$:

(**Q2**) *Is the semigroup* $\mathcal{A}(\mathbb{N}^{\gamma_n})$ *normal, i.e., does it coincide with* $\mathcal{A}(\mathbb{R}^{\gamma_n}_{\geq 0}) \cap \mathbb{Z}^{2^n}$ *?*

In Section 5 we study the toric ideal [3] of $\mathcal{A}$ in a polynomial ring in γ_n unknowns, and we examine how it differs from the subideal generated by the binomials

$$(\mathbf{SG}''') \qquad [i \perp\!\!\!\perp j \,|K \cup \ell] \cdot [i \perp\!\!\!\perp \ell \,|K] \;-\; [i \perp\!\!\!\perp j \,|K] \cdot [i \perp\!\!\!\perp \ell \,|K \cup j]. \qquad .$$

Proposition 7.5.1 describes the primary decomposition of this binomial ideal for $n = 4$. We also discuss the problem of deriving the full Markov basis from (**SG**‴).

7.2. A non-submodular simplicial semigraphoid

Let $n = 4$ and consider the 4-dimensional cube C_4 and the 3-dimensional permutohedron Π_3. Each hexagon on Π_3 corresponds to one of the eight facets of C_4. Each facet specifies three semigraphoid axioms, written additively as in (**SG**″):

$$\begin{array}{lrcll}
 & [\![1\perp\!\!\!\perp 2|\emptyset]\!] + [2\perp\!\!\!\perp 3|1] & = & [2\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 2|3] & \Longleftarrow \\
(*,*,*,0) & [1\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 2|3] & = & [\![1\perp\!\!\!\perp 2|\emptyset]\!] + [1\perp\!\!\!\perp 3|2] & \\
 & [1\perp\!\!\!\perp 3|\emptyset] + [2\perp\!\!\!\perp 3|1] & = & [2\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 3|2] & \\
 & & & & \\
 & [\![1\perp\!\!\!\perp 2|\emptyset]\!] + [2\perp\!\!\!\perp 4|1] & = & [2\perp\!\!\!\perp 4|\emptyset] + [1\perp\!\!\!\perp 2|4] & \\
(*,*,0,*) & [\![1\perp\!\!\!\perp 2|\emptyset]\!] + [1\perp\!\!\!\perp 4|2] & = & [1\perp\!\!\!\perp 4|\emptyset] + [1\perp\!\!\!\perp 2|4] & \\
 & [1\perp\!\!\!\perp 4|\emptyset] + [2\perp\!\!\!\perp 4|1] & = & [2\perp\!\!\!\perp 4|\emptyset] + [1\perp\!\!\!\perp 4|2] & \\
 & & & & \\
 & [1\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 4|3] & = & [1\perp\!\!\!\perp 4|\emptyset] + [1\perp\!\!\!\perp 3|4] & \\
(*,0,*,*) & [\![3\perp\!\!\!\perp 4|\emptyset]\!] + [1\perp\!\!\!\perp 3|4] & = & [1\perp\!\!\!\perp 3|\emptyset] + [3\perp\!\!\!\perp 4|1] & \\
 & [\![3\perp\!\!\!\perp 4|\emptyset]\!] + [1\perp\!\!\!\perp 4|3] & = & [1\perp\!\!\!\perp 4|\emptyset] + [3\perp\!\!\!\perp 4|1] & \Longleftarrow
\end{array}$$

$$
\begin{array}{lrcll}
 & [2\perp\!\!\!\perp 3|\emptyset] + [3\perp\!\!\!\perp 4|2] & = & [\![3\perp\!\!\!\perp 4|\emptyset]\!] + [2\perp\!\!\!\perp 3|4] & \\
(0,*,*,*) & [2\perp\!\!\!\perp 4|\emptyset] + [2\perp\!\!\!\perp 3|4] & = & [2\perp\!\!\!\perp 3|\emptyset] + [2\perp\!\!\!\perp 4|3] & \\
 & [\![3\perp\!\!\!\perp 4|\emptyset]\!] + [2\perp\!\!\!\perp 4|3] & = & [2\perp\!\!\!\perp 4|\emptyset] + [3\perp\!\!\!\perp 4|2] & \\
 & & & & \\
 & [3\perp\!\!\!\perp 4|1] + [\![2\perp\!\!\!\perp 3|14]\!] & = & [2\perp\!\!\!\perp 3|1] + [3\perp\!\!\!\perp 4|12] & \Longleftarrow \\
(*,*,*,1) & [2\perp\!\!\!\perp 4|1] + [\![2\perp\!\!\!\perp 3|14]\!] & = & [2\perp\!\!\!\perp 3|1] + [2\perp\!\!\!\perp 4|13] & \\
 & [2\perp\!\!\!\perp 4|1] + [3\perp\!\!\!\perp 4|12] & = & [3\perp\!\!\!\perp 4|1] + [2\perp\!\!\!\perp 4|13] & \\
 & & & & \\
 & [1\perp\!\!\!\perp 3|2] + [3\perp\!\!\!\perp 4|12] & = & [3\perp\!\!\!\perp 4|2] + [1\perp\!\!\!\perp 3|24] & \\
(*,*,1,*) & [1\perp\!\!\!\perp 3|2] + [\![1\perp\!\!\!\perp 4|23]\!] & = & [1\perp\!\!\!\perp 4|2] + [1\perp\!\!\!\perp 3|24] & \\
 & [3\perp\!\!\!\perp 4|2] + [\![1\perp\!\!\!\perp 4|23]\!] & = & [1\perp\!\!\!\perp 4|2] + [3\perp\!\!\!\perp 4|12] & \\
 & & & & \\
 & [1\perp\!\!\!\perp 2|3] + [\![1\perp\!\!\!\perp 4|23]\!] & = & [1\perp\!\!\!\perp 4|3] + [1\perp\!\!\!\perp 2|34] & \Longleftarrow \\
(*,1,*,*) & [1\perp\!\!\!\perp 4|3] + [2\perp\!\!\!\perp 4|13] & = & [2\perp\!\!\!\perp 4|3] + [\![1\perp\!\!\!\perp 4|23]\!] & \\
 & [1\perp\!\!\!\perp 2|3] + [2\perp\!\!\!\perp 4|13] & = & [2\perp\!\!\!\perp 4|3] + [1\perp\!\!\!\perp 2|34] & \\
 & & & & \\
 & [1\perp\!\!\!\perp 3|4] + [\![2\perp\!\!\!\perp 3|14]\!] & = & [2\perp\!\!\!\perp 3|4] + [1\perp\!\!\!\perp 3|24] & \\
(1,*,*,*) & [1\perp\!\!\!\perp 2|4] + [1\perp\!\!\!\perp 3|24] & = & [1\perp\!\!\!\perp 3|4] + [1\perp\!\!\!\perp 2|34] & \\
 & [1\perp\!\!\!\perp 2|4] + [\![2\perp\!\!\!\perp 3|14]\!] & = & [2\perp\!\!\!\perp 3|4] + [1\perp\!\!\!\perp 2|34]. &
\end{array}
$$

This is a system of 24 equations in $\gamma_4 = 24$ formal symbols $[i \perp\!\!\!\perp j \,|\, K]$.

A semigraphoid is a solution to these equations over the semiring $\{0, +\}$. More precisely, given such a solution vector in $\{0, +\}^{24}$, the semigraphoid $\mathcal{M}$ consists of all coordinates $[i \perp\!\!\!\perp j \,|\, K]$ that have the value 0. There are 26424 such semigraphoids. They form a sublattice of the Boolean lattice $\{0, +\}^{24}$, with $+ < 0$. Question (**Q1**) concerns the coatoms of this lattice. But let us first resolve Question (**Q3**).

We consider the following collection of CI statements:

$$\mathcal{M} \;=\; \big\{\, [\![2 \perp\!\!\!\perp 3 \,|\, 14]\!],\, [\![1 \perp\!\!\!\perp 4 \,|\, 23]\!],\, [\![1 \perp\!\!\!\perp 2 \,|\, \emptyset]\!],\, [\![3 \perp\!\!\!\perp 4 \,|\, \emptyset]\!] \,\big\}. \tag{7.2}$$

These four symbols are highlighted in the 24 equations above by the use of double brackets $[\![\cdots]\!]$. Each equation (individually) can be solved among the positive reals after these four symbols have been set to zero, or equivalently they can be solved as a system over $\{0, +\}$. This shows that $\mathcal{M}$ is a semigraphoid.

The semigraphoid $\mathcal{M}$ is represented geometrically by the three-dimensional polytope in Figure 1. This polytope is *simple*, i.e., each of the 16 vertices is adjacent to three other vertices. The eight vertices whose labels include three bars (such as 4|2|1|3) correspond to unique permutations in S_4 (namely the permutation 4213), while the eight vertices whose labels have two bars (such as 4|1|23) correspond to pairs of permutations in S_4 (namely 4123 and 4132). This partition of S_4 into eight singletons and eight pairs is the convex rank test of $\mathcal{M}$. The normal fan of the polytope in Figure 1 is a simplicial fan which is combinatorially (but not geometrically) isomorphic to a fan that coarsens the hyperplane arrangement of S_4.

Proposition 7.2.1. *The simplicial semigraphoid $\mathcal{M}$ is not submodular.*

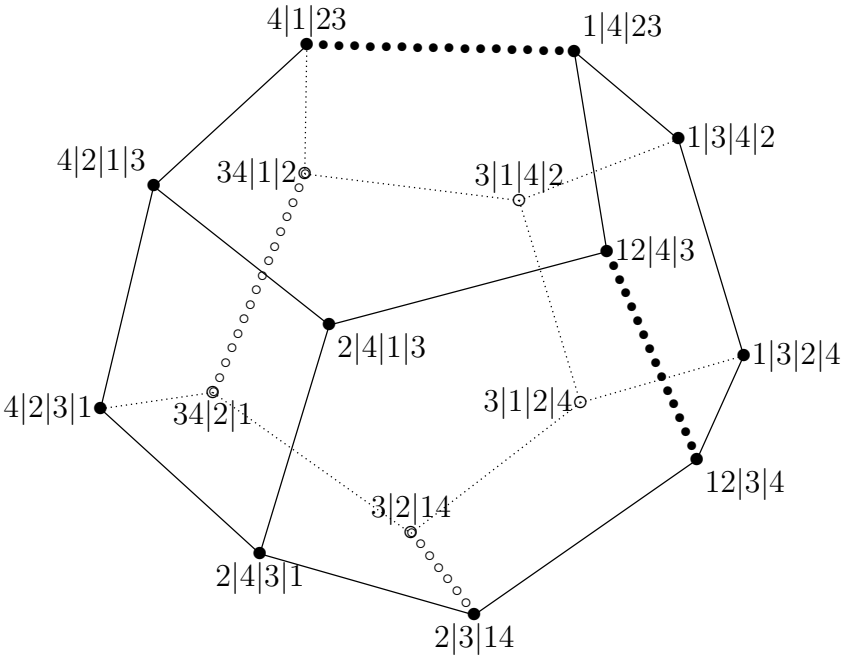


Figure 7.1.: A simple 3-dimensional polytope with 16 vertices and 10 facets

Proof. Suppose that $\mathcal{M}$ were submodular. Then the above equations have a solution in $(\mathbb{R}_{\geq 0})^{24}$ whose coordinates in $\mathcal{M}$ are zero and whose other 20 coordinates are positive. The four equations marked by an "$\Longleftarrow$" give the following four equations:

$$\begin{array}{rcl} [2\perp\!\!\!\perp 3|1] & = & [2\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 2|3] \\ [1\perp\!\!\!\perp 4|3] & = & [1\perp\!\!\!\perp 4|\emptyset] + [3\perp\!\!\!\perp 4|1] \\ [3\perp\!\!\!\perp 4|1] & = & [2\perp\!\!\!\perp 3|1] + [3\perp\!\!\!\perp 4|12] \\ [1\perp\!\!\!\perp 2|3] & = & [1\perp\!\!\!\perp 4|3] + [1\perp\!\!\!\perp 2|34]. \end{array}$$

Adding the left hand sides and the right hand sides of the four equations yields

$$[2\perp\!\!\!\perp 3|\emptyset] + [1\perp\!\!\!\perp 4|\emptyset] + [3\perp\!\!\!\perp 4|12] + [1\perp\!\!\!\perp 2|34] \quad = \quad 0.$$

This contradicts the assumption that these four values are strictly positive. □

The set of all non-negative solutions to the 24 equations is an 11-dimensional cone in $(\mathbb{R}_{\geq 0})^{24}$. This cone is isomorphic to the 16-dimensional cone of submodular functions on $2^{[4]}$, modulo its 5-dimensional lineality space. Its 22108 faces are in bijection with the submodular semigraphoids, or, equivalently, with the generalized permutohedra for $n = 4$. In addition to these, there are 4316 semigraphoids that are not submodular. Each of the latter can be represented by a polytope of dimension ≤ 3 as in Figure 1. These polytopes have the combinatorial properties of generalized permutohedra, but they cannot be realized as Minkowski summands of Π_3. For example, see [42, Figure 5] for a polytope that depicts Studený's example of a semigraphoid that is not submodular (see [54] and [80, Section 2.2.4]).

We now give a classification of non-submodular semigraphoids for $n = 4$ and $|\mathcal{M}$ small. All simplicial examples are coarsenings (up to relabeling) of the particular semigraphoid $\mathcal{M}$ in Proposition 7.2.1. The following table lists the number of semigraphoids classified by number of CI statements, their type, and whether they are simplicial. Here, the *type* of a semigraphoid is the triple (m_0, m_1, m_2) where m_t is the number of CI statements $[i \perp\!\!\!\perp j \,|\, K]$ in $\mathcal{M}$ such that $|K| = m_t$.

$\|\mathcal{M}\|$	type	non-simplicial	simplicial	total
3	(0 , 3 , 0)	8	0	8
4	(0 , 4 , 0)	78	0	78
4	(1 , 2 , 1)	30	0	30
4	(2 , 0 , 2)	0	6	6
5	(0 , 5 , 0)	300	0	300
5	(1 , 2 , 2)	30	0	30
5	(1 , 3 , 1)	84	0	84
5	(2 , 0 , 3)	12	12	24
5	(2 , 2 , 1)	30	0	30
5	(3 , 0 , 2)	24	0	24
6	(0 , 6 , 0)	604	0	604
6	(1 , 3 , 2)	84	0	84
6	(1 , 4 , 1)	78	0	78
6	(2 , 0 , 4)	30	3	33
6	(2 , 2 , 2)	30	0	30
6	(2 , 3 , 1)	84	0	84
6	(3 , 0 , 3)	74	12	96
6	(4 , 0 , 2)	30	3	33
7	(0 , 7 , 0)	684	0	684
7	(1 , 4 , 2)	78	0	78
7	(1 , 5 , 1)	24	0	24
7	(2 , 0 , 5)	18	0	18
7	(2 , 3 , 2)	84	0	84
7	(2 , 4 , 1)	78	0	78
7	(3 , 0 , 4)	132	0	132
7	(4 , 0 , 3)	132	0	132
7	(5 , 0 , 2)	18	0	18
8	(0 , 8 , 0)	450	0	450
8	(1 , 5 , 2)	24	0	24
8	(2 , 0 , 6)	3	0	3
8	(2 , 4 , 2)	48	0	48

$\|\mathcal{M}\|$	type	non-simplicial	simplicial	total
8	(2 , 5 , 1)	24	0	24
8	(3 , 0 , 5)	72	0	72
8	(4 , 0 , 4)	174	0	174
8	(5 , 0 , 3)	72	0	72
8	(6 , 0 , 2)	3	0	3
9	(0 , 9 , 0)	212	0	212
9	(3 , 0 , 6)	12	0	12
9	(4 , 0 , 5)	84	0	84
9	(5 , 0 , 4)	84	0	84
9	(6 , 0 , 3)	12	0	12
10	(0 , 10 , 0)	60	0	60
10	(4 , 0 , 6)	15	0	15
10	(5 , 0 , 5)	24	0	24
10	(6 , 0 , 4)	15	0	15
11	(0 , 11 , 0)	12	0	12
11	(5 , 0 , 6)	6	0	6
11	(6 , 0 , 5)	6	0	6

7.3. A non-submodular coarsest semigraphoid

We now consider the case $n = 5$. There are $\gamma_5 = 80$ CI statements, one for each two-dimensional face of the 5-cube C_5. There are 120 semigraphoid axioms (**SG″**), three for each of the 40 three-dimensional faces of C_5, listed as additive equations in the Appendix. The semigraphoids are the solutions of these equations over $\{0, +\}^{80}$. These solutions include the all-zero vector **0** which represents the semigraphoid that consists of all 80 CI statements, and which is the maximal element in the lattice of semigraphoids. A semigraphoid is said to be **coarsest** if it is maximal among non-**0** semigraphoids. Geometrically, such a semigraphoid corresponds to a fan which coarsens the S_5-arrangement but cannot be coarsened to a non-trivial fan.

We now present the counterexample which answers question (**Q1**). The constructions make use of the identification of semigraphoids with convex rank tests that was derived in [54]. Let Γ denote the partition of the symmetric group S_5 into fourteen classes as follows. There are eight classes containing 12 permutations each:

$$\begin{array}{cccc} 15|234 & 234|15 & 123|45 & 235|14 \\ 124|35 & 245|13 & 134|25 & 345|12. \end{array}$$

And there are six classes containing four permutations each:

$$\begin{array}{ccc} 12|5|34 & 25|1|34 & 13|5|24 \\ 35|1|24 & 14|5|23 & 45|1|23. \end{array}$$

Here $15|234$ denotes the class of all permutations $ijklm$ with $\{i, j\} = \{1, 5\}$ and $\{k, l, m\} = \{2, 3, 4\}$. Similarly, $45|1|23$ denotes the class of all permutations $ijklm$

with $\{i,j\} = \{4,5\}$, $k = 1$, and $\{l,m\} = \{2,3\}$. Clearly, Γ is a pre-convex rank test, as each of the 14 classes is the set of all linear extensions of a poset on $[5] = \{1,2,3,4,5\}$. Note that the stabilizer of the pre-convex rank test Γ in S_5 has order 12, because Γ is fixed under permutations of $\{1,5\}$ and permutations of $\{2,3,4\}$. The 14 classes of Γ are represented by the 14 vertices of the polytope in Figure 2.

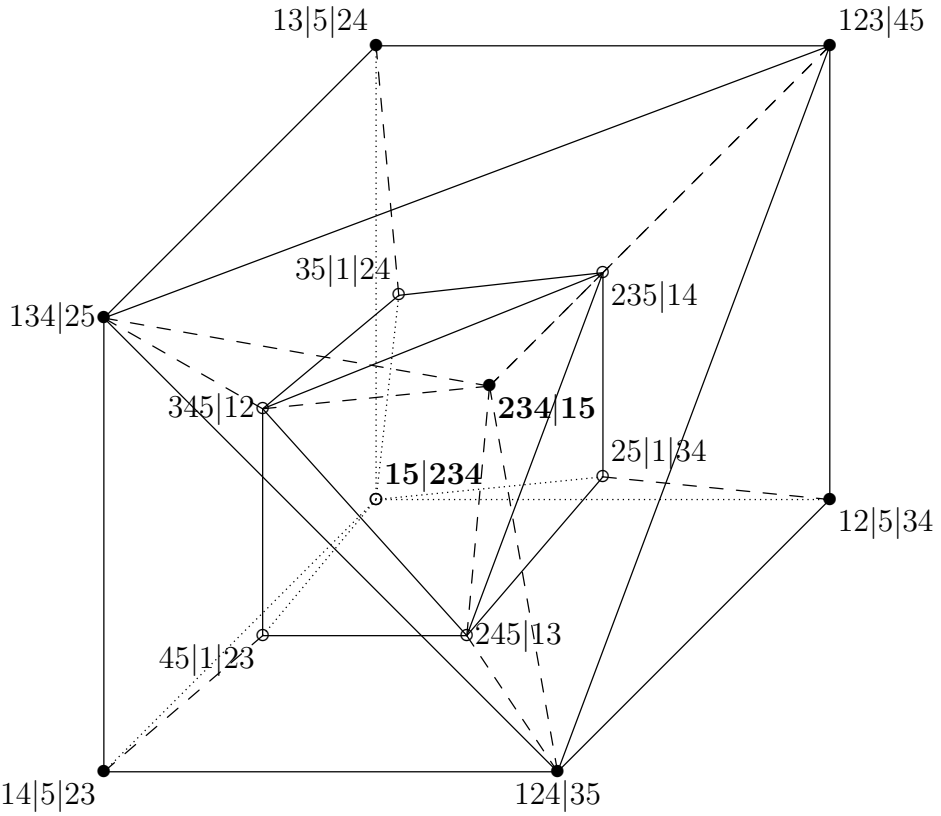


Figure 7.2.: Schlegel diagram of a 4-dimensional polytope with 10 facets

Each pair of adjacent permutations in a given class of Γ specifies a CI statement. For instance, the four-element class $45|1|23$ specifies the two CI statements $[\![4 \perp\!\!\!\perp 5|\emptyset]\!]$ and $[\![2 \perp\!\!\!\perp 3|145]\!]$, while the 12-element class $15|234$ specifies the seven CI statements

$$[\![1 \perp\!\!\!\perp 5|\emptyset]\!], [\![2 \perp\!\!\!\perp 3|15]\!], [\![2 \perp\!\!\!\perp 3|145]\!], [\![2 \perp\!\!\!\perp 4|15]\!], [\![2 \perp\!\!\!\perp 4|135]\!], [\![3 \perp\!\!\!\perp 4|15]\!], [\![3 \perp\!\!\!\perp 4|125]\!].$$

Altogether, we obtain 44 CI statements $[\![\cdot \mid \cdot]\!]$ from the 14 classes, and we identify the pre-convex rank test Γ with this set of 44 CI statements. We now prove:

Theorem 7.3.1. *Γ is a coarsest convex rank test which is not submodular.*

Proof. To establish this theorem, we must prove the following three claims:

- Γ is a convex rank test, i.e. it satisfies the semigraphoid axioms (**SG**).
- There is no proper convex rank test which is coarser than Γ.

- The convex rank test Γ is not submodular.

We shall prove all three statements at once, by examining the semigraphoid equations (**SG″**). As in Section 2, the 44 symbols in Γ are denoted with double brackets $[\![\cdot|\cdot]\!]$, while the 36 symbols not in Γ are denoted with brackets $[\cdot|\cdot]$. With this distinction between brackets, there are four symmetry types of semigraphoid equations that involve the 36 positive unknowns $[\cdot|\cdot]$. The full list is given in the Appendix:

$$\begin{array}{lrcl}
\text{Type I} & [3\perp\!\!\!\perp 5|12] + [\![3\perp\!\!\!\perp 4|125]\!] & = & [3\perp\!\!\!\perp 4|12] + [\![3\perp\!\!\!\perp 5|124]\!] \\
\text{Type II} & [1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 3|25] & = & [\![1\perp\!\!\!\perp 3|2]\!] + [1\perp\!\!\!\perp 5|23] \\
\text{Type III} & [4\perp\!\!\!\perp 5|1] + [2\perp\!\!\!\perp 5|14] & = & [2\perp\!\!\!\perp 5|1] + [4\perp\!\!\!\perp 5|12] \\
\text{Type IV} & [1\perp\!\!\!\perp 2|5] + [\![2\perp\!\!\!\perp 3|15]\!] & = & [\![2\perp\!\!\!\perp 3|5]\!] + [1\perp\!\!\!\perp 2|35]
\end{array}$$

After setting the 44 unknowns $[\![\cdot|\cdot]\!]$ to zero, we are left with 120 equations in the 36 strictly positive unknowns. For instance, the first three types give

$$\begin{array}{lrcl}
\text{Type I} & [3\perp\!\!\!\perp 5|12] & = & [3\perp\!\!\!\perp 4|12] \\
\text{Type II} & [1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 3|25] & = & [1\perp\!\!\!\perp 5|23] \\
\text{Type III} & [4\perp\!\!\!\perp 5|1] + [2\perp\!\!\!\perp 5|14] & = & [2\perp\!\!\!\perp 5|1] + [4\perp\!\!\!\perp 5|12]
\end{array}$$

The axiom (**SG″**) merely requires that each of these equations is *individually* solvable. This is obviously the case. Hence Γ is a semigraphoid.

The 78 equations of Type I listed in the Appendix imply that all 36 positive unknowns must be equal. So, if another CI statement is added to the semigraphoid Γ, then all others must be added in order for (**SG**) to remain valid. This proves the second claim that Γ is a coarsest convex rank test.

Given that the 36 unknowns $[\cdot|\cdot]$ must be equal, the 12 Type II equations imply that their common value is zero, contradicting the requirement that they be positive. Hence the 120 orginal equations *altogether* have no non-negative real solution that is consistent with Γ. This proves the third claim that Γ is not submodular. □

Every semigraphoid for $n = 5$ corresponds to a 4-dimensional fan. Intersecting this fan with a sphere around the origin, we obtain a polyhedral cell decomposition of the 3-dimensional sphere. We do not know whether each of these 3-spheres can be realized as the boundary of a 4-dimensional polytope. However, using [92, §5], every semigraphoid can be represented by a 3-dimensional diagram as in Figure 2.

For the specific semigraphoid Γ of Theorem 7.3.1, the diagram in Figure 2 is indeed the boundary of a 4-polytope with f-vector $(14, 36, 32, 10)$. The following coordinates for this polytope were found by a direct calculation, using the techniques described in [6]. Each of the following ten row vectors represents a facet of the polytope:

```
POINTS
 1      1/4      0      0      0      0
 1       0       1      0      0      0
 1       0       0      1      0      0
 1       0       0      0      1      0
```

```
1       0          0        0        0        1
1     -1/4        1/4      1/4      5/4      1/4
1    280/893   -280/893   25/893     0      28/893
1     1/57       1/57     -1/57    17/19     2/57
1      1          1        0       -5        1
1     2/37       20/37    1/37     10/37    -2/37
```

For instance, the last row represents the facet-defining inequality

$$\frac{2}{37} \cdot x_1 + \frac{20}{37} \cdot x_2 + \frac{1}{37} \cdot x_3 + \frac{10}{37} \cdot x_4 - \frac{2}{37} \cdot x_5 \ \leq \ 1.$$

Here, we are considering the vectors $(x_1, x_2, x_3, x_4, x_5)$ to be elements in the quotient of $\mathbb{R}^5$ modulo the one-dimensional linear subspace spanned by $(4, 1, 1, 1, 1)$. The format is that of the software `polymake` [30]. If the above eleven lines are put in a file named `mypolytope` then the following command in `polymake` will verify that this polytope does indeed have the combinatorial structure displayed in Figure 2:

```
polymake mypolytope F_VECTOR VERTICES_IN_FACETS
```

The 10 facets of the 4-polytope correspond to the facets of the 5-cube, and they comprise all classes of permutations in S_5 in which the first or last coordinate is fixed. The facets corresponding to permutations with 1 *or* 5 in the *first* coordinate have seven vertices, twelve edges, and eight 2-faces. The facets corresponding to permutations with 2, 3 *or* 4 *first* have seven vertices, 13 edges, and eight 2-faces. The facets for 1 *or* 5 *last* are tetrahedra. The facets for 2, 3 *or* 4 *last* are cubes in which one edge has been contracted; they have seven vertices and 11 edges.

7.4. The semigraphoid semigroup is not normal

Continuing to assume $n = 5$, we now consider the linear map $\mathcal{A}$ in the Introduction. It maps the free abelian group $\mathbb{Z}^{80}$ spanned by the CI statements to the free abelian group $\mathbb{Z}^{32}$ with basis $\{e_K : K \subseteq [5]\}$ as specified in (7.1). The matrix representing $\mathcal{A}$ has 32 rows and 80 columns; each column has four non-zero entries: two $+1$'s and two -1's. The rank of $\mathcal{A}$ is 26. The *semigraphoid semigroup* is $\mathcal{A}(\mathbb{N}^{80})$, the non-negative integer span of the columns of this 32×80-matrix. This is a subsemigroup of $\mathbb{Z}^{32}$. Equivalently, the semigraphoid semigroup is the affine semigroup with 80 generators and 120 relations (given in the Appendix). Note that the polyhedral cone dual to the semigraphoid semigroup is the cone of submodular functions.

In the language of [80], the vectors in $\mathbb{Z}^{32}$ are called *imsets*, the columns of $\mathcal{A}$ are *elementary imsets*, and the elements of $\mathcal{A}(\mathbb{N}^{80})$ are *combinatorial imsets*. A *structural imset* is a lattice point which lies in the polyhedral cone spanned by the elementary imsets. Studený's question (**Q2**) whether each structural imset is combinatorial translates into the question whether the semigroup $\mathcal{A}(\mathbb{N}^{80})$ is normal.

Theorem 7.4.1. *The semigraphoid semigroup is not normal for $n = 5$.*

Proof. Consider the following element in the free abelian group $\mathbb{Z}^{80}$:

$$\begin{array}{c}[1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 4|3] + [2\perp\!\!\!\perp 3|4] + [2\perp\!\!\!\perp 3|5] + [3\perp\!\!\!\perp 4|12] \\ +[2\perp\!\!\!\perp 5|13] + [1\perp\!\!\!\perp 2|45] + [1\perp\!\!\!\perp 3|45] + [4\perp\!\!\!\perp 5|23] - [2\perp\!\!\!\perp 3|45].\end{array} \tag{7.3}$$

The image of this element under the map $\mathcal{A} : \mathbb{Z}^{80} \to \mathbb{Z}^{32}$ is the imset

$$\begin{array}{rl}\mathbf{b} := & -e_2 - e_3 - e_4 - e_5 - e_{23} + e_{24} + 2e_{25} + 2e_{34} + e_{35} - e_{45} + 2e_{123} \\ & +e_{124} - e_{125} - e_{134} + e_{135} + 2e_{145} - e_{1234} - e_{1235} - e_{1245} - e_{1345}.\end{array} \tag{7.4}$$

The imset $\mathbf{b}$ is structural because $2 \cdot \mathbf{b}$ is a combinatorial imset. It is the image of

$$\begin{array}{l}[4\perp\!\!\!\perp 5|2] + [4\perp\!\!\!\perp 5|3] + [1\perp\!\!\!\perp 3|4] + [1\perp\!\!\!\perp 2|5] + [2\perp\!\!\!\perp 5|14] + [3\perp\!\!\!\perp 4|15] \\ \quad +[1\perp\!\!\!\perp 4|23] + [1\perp\!\!\!\perp 5|23] + [1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 4|3] + [2\perp\!\!\!\perp 3|4] \\ \quad +[2\perp\!\!\!\perp 3|5] + [3\perp\!\!\!\perp 4|12] + [2\perp\!\!\!\perp 5|13] + [1\perp\!\!\!\perp 2|45] + [1\perp\!\!\!\perp 3|45] \qquad \in \mathbb{N}^{80}\end{array} \tag{7.5}$$

under the linear map $\mathcal{A}$.

Suppose that $\mathbf{b}$ were a combinatorial imset. Then there exists $\mathbf{x} \in \mathbb{N}^{80}$ such that $\mathcal{A} \cdot \mathbf{x} = \mathbf{b}$. We write $\mathbf{x} = \sum_i [a_i \perp\!\!\!\perp b_i | K_i]$, where we allow repetition in the sum. In any elementary imset, the basis vector $e_\emptyset$ occurs with coefficient -1 or 0, and the basis vector e_{12345} occurs with coefficient -1 or 0. However, neither $e_\emptyset$ nor e_{12345} appears in the imset $\mathbf{b}$, so we conclude that $|K_i| = 1$ or $|K_i| = 2$ for all terms $[a_i \perp\!\!\!\perp b_i | K_i]$ in the representation of $\mathbf{x}$. The first four terms $-e_2 - e_3 - e_4 - e_5$ in $\mathbf{b}$ imply that $\mathbf{x}$ has precisely four terms $[a_i \perp\!\!\!\perp b_i | K_i]$ with $|K_i| = 1$, and the terms $-e_{1234} - e_{1235} - e_{1245} - e_{1345}$ imply that $\mathbf{x}$ has precisely four terms with $|K_i| = 2$.

Each of the eight terms in $\mathbf{x}$ evaluates to an alternating sum of 4 terms under the map $\mathcal{A}$. Some cancelation occurs among the resulting 32 terms. Prior to that cancelation, the imset had been written as the sum of two subsums, $\mathbf{b} = \mathcal{A} \cdot \mathbf{x} =$

$$-e_2 - e_3 - e_4 - e_5 + e_{24} + 2e_{25} + 2e_{34} + e_{35} + e_{A_1} + e_{A_2} - e_{125} - e_{134} - e_{B_1} - e_{B_2}$$

$$-e_{23} - e_{45} - e_{A_1} - e_{A_2} + 2e_{123} + e_{124} + e_{135} + 2e_{145} + e_{B_1} + e_{B_2} - e_{1234} - e_{1235} - e_{1245} - e_{1345},$$

where $|A_1| = |A_2| = 2$ and $|B_1| = |B_2| = 3$. The first line is the sum of the four elementary imsets $\mathcal{A}([a_i \perp\!\!\!\perp b_i | K_i])$ with $|K_i| = 1$, and the second line is the sum of the four elementary imsets with $|K_i| = 2$. A contradiction will arise when we try to determine the unknown pairs A_1 and A_2. The term $-e_{125}$ in the first line must come from $K_i = \{2\}$ or $K_i = \{5\}$. This implies that either $\{1, 2\}$ or $\{1, 5\}$ is in $A_* = \{A_1, A_2\}$. Similarly, the term $-e_{134}$ shows that either $\{1, 3\}$ or $\{1, 4\}$ is in A_*. Now consider the second line. The presence of the term $2e_{123}$ implies that $\{1, 2\}$ or $\{1, 3\}$ is in A_*, and the term $2e_{145}$ implies that $\{1, 4\}$ or $\{1, 5\}$ is in A_*. The term e_{124} shows that $\{1, 2\}$, $\{1, 4\}$, or $\{2, 4\}$ is in A_*, and, finally, the term e_{135} shows that $\{1, 3\}$, $\{1, 5\}$, or $\{3, 5\}$ is in A_*. However, no such pair of pairs A_* satisfies these six restrictions. This proves that $\mathbf{b}$ is not a combinatorial imset. □

The main point of the above proof was to show that the linear system $\mathcal{A}\cdot\mathbf{x} = \mathbf{b}$ has no solution with non-negative *integer* coordinates. This can also be verified automatically using integer programming software. In fact, using such software we found that $\mathcal{A}\cdot\mathbf{x} = \mathbf{b}$ has only one solution with non-negative *real* coordinates, namely, that unique solution $\mathbf{x} \in (\mathbb{R}_{\geq 0})^{80}$ is the expression in (7.5) scaled by $1/2$.

The reader might now inquire how the imset $\mathbf{b}$ was found. There are several algorithms that test whether a given affine semigroup is normal, including one recently proposed by Takemura, Yoshida and the first author [41], and the method of Bruns and Koch [14] which is implemented in their software `normaliz`.

The original attempts to apply these methods directly to the 32×80-matrix $\mathcal{A}$ were unsuccessful. Instead we succeeded by partially computing a *Markov basis* for the matrix $\mathcal{A}$ using the software `4ti2` [39]. The imset $\mathbf{b}$ was found by inspecting the partial results produced by `4ti2`. We explain the details in the next section.

7.5. Computations in toric algebra

Let $\mathbb{Q}[\mathrm{CI}_n]$ denote the polynomial ring over the field of rational numbers $\mathbb{Q}$ generated by the symbols $[i \perp\!\!\!\perp j \,|\, K]$. Thus $\mathbb{Q}[\mathrm{CI}_n]$ is a polynomial ring in γ_n unknowns, one for each 2-face of the n-cube C_n. We write $\prod \mathrm{CI}_n$ for the product of all the unknowns. We define the *semigraphoid ideal* to be the ideal $I_{\mathbf{SG}}$ generated by the binomials in ($\mathbf{SG}'''$). Thus the generators of $I_{\mathbf{SG}}$ represent the semigraphoid axioms. Following [53, §7], we introduce the *toric ideal* $I_{\mathcal{A}}$ which is obtained from $I_{\mathbf{SG}}$ by saturation:

$$I_{\mathcal{A}} \quad := \quad \bigl(I_{\mathbf{SG}} \,:\, (\prod \mathrm{CI}_n)^{\infty}\bigr). \tag{7.6}$$

The binomials in $I_{\mathcal{A}}$ represent the vectors in the kernel of the linear map $\mathcal{A} : \mathbb{Z}^{\gamma_n} \to \mathbb{Z}^{2^n}$. A minimal set of binomials which generates $I_{\mathcal{A}}$ is said to be a *Markov basis* for the matrix $\mathcal{A}$. See [27] for a discussion of Markov bases in the context of statistics.

Let us illustrate these concepts for $n = 3$. The polynomial ring $\mathbb{Q}[\mathrm{CI}_3]$ has six unknowns, one for each facet of the 3-cube. They are the entries of the 2×3-matrix

$$\begin{pmatrix} [1 \perp\!\!\!\perp 2|\emptyset] & [1 \perp\!\!\!\perp 3|\emptyset] & [2 \perp\!\!\!\perp 3|\emptyset] \\ [1 \perp\!\!\!\perp 2|3] & [1 \perp\!\!\!\perp 3|2] & [2 \perp\!\!\!\perp 3|1] \end{pmatrix}. \tag{7.7}$$

The semigraphoid ideal $I_{\mathbf{SG}}$ is generated by the three 2×2-minors of the matrix (7.7). This is a prime ideal of codimension 2 and degree 3, and hence we have $I_{\mathbf{SG}} = I_{\mathcal{A}}$. Here the Markov basis for $\mathcal{A}$ consists precisely of the three semigraphoid axioms.

We next consider the case $n = 4$. The polynomial ring $\mathbb{Q}[\mathrm{CI}_4]$ has 24 unknowns, one for each 2-face of the 4-cube. They are the entries of eight 2×3-matrices as in (7.7), one for each of the eight facets of the 4-cube. Thus the semigraphoid ideal $I_{\mathbf{SG}}$ is generated by 24 quadrics, one for each of the 24 axioms ($\mathbf{SG}''$) in the list given in Section 2. For instance, the last axiom in that list translates into the quadratic binomial $[1 \perp\!\!\!\perp 2|4] \cdot [2 \perp\!\!\!\perp 3|14] - [2 \perp\!\!\!\perp 3|4] \cdot [1 \perp\!\!\!\perp 2|34]$, which is one of the 24 generators of $I_{\mathbf{SG}}$. Using the software `Macaulay2` [34] we derived the following result:

Proposition 7.5.1. *The semigraphoid ideal $I_{\mathbf{SG}}$ is a radical ideal which is the intersection of the toric ideal $I_{\mathcal{A}}$ and 17 additional associated monomial prime ideals.*

Before discussing this prime decomposition in detail, let us make a few general remarks. We wish to argue that toric algebra and algebraic geometry provide useful algorithmic tools for the research directions presented in [80]. For any ideal I of $\mathbb{Q}[\mathrm{CI}_n]$ and any subset Ω of the complex affine space $\mathbb{C}^{\gamma_n}$, the *variety* $V_\Omega(I)$ is defined as the set of all vectors in Ω which are common zeros of all the polynomials in I. Then $V_{\mathbb{C}}(I_{\mathbf{SG}})$ is a complex variety, reducible for $n \geq 4$, one of whose irreducible components is the complex toric variety $V_{\mathbb{C}}(I_{\mathcal{A}})$. Inside this toric variety are the real toric variety $V_{\mathbb{R}}(I_{\mathcal{A}})$. Its non-negative part $V_{\mathbb{R}_{\geq 0}}(I_{\mathcal{A}})$ is homeomorphic to the cone spanned by the elementary imsets. The next result shows that the semigraphoids are precisely the points on these varieties whose coordinates are 0 or 1.

Theorem 7.5.2. *The semigraphoids on $[n]$ are in bijection with the points in $V_{\{0,1\}}(I_{\mathbf{SG}})$. The submodular semigraphoids on $[n]$ are in bijection with the points in $V_{\{0,1\}}(I_{\mathcal{A}})$.*

Proof. We replace the additive semiring $\{0, +\}$ with the multiplicative semiring $\{1, 0\}$. This translates from the additive notation ($\mathbf{SG}''$) to the multiplicative notation ($\mathbf{SG}'''$). With this translation, the first statement in Theorem 7.5.2 is obvious.

The second statement is less obvious and is based on the geometry of toric varieties. Specifically, we shall use the characterization of *facial* index sets which is developed in [32]. If we consider the specific $2^n \times \gamma_n$-matrix $\mathcal{A}$ then the role of the set $\{1, \ldots, m\}$ in [32] is played by the set of CI statements, and a subset of CI statements is facial for $\mathcal{A}$ if and only if it is submodular semigraphoid. With this observation, the second assertion follows from Lemma A.2 in the Appendix of [32]. □

Using Theorem 7.5.2, we can study semigraphoids by studying the zero-dimensional ideals obtained by adding $\langle\, x^2 - x \,:\, x \in \mathrm{CI}_n \,\rangle$ to the ideal $I_{\mathbf{SG}}$ or $I_{\mathcal{A}}$. For instance, with the command `degree` in `Macaulay2` [34], it takes only a few seconds to compute

$$\#V_{\{0,1\}}(I_{\mathbf{SG}}) \,=\, 26424 \qquad \text{and} \qquad \#V_{\{0,1\}}(I_{\mathcal{A}}) \,=\, 22108. \tag{7.8}$$

The difference between these numbers is explained geometrically by the prime decomposition in Proposition 7.5.1, which we shall now describe in explicit terms.

The 17 associated monomial primes of $I_{\mathbf{SG}}$ come in three symmetry classes. First there are two primes of codimension 12. A representative is the ideal

$$\left\langle \begin{array}{c} [1 \perp\!\!\!\perp 2|\emptyset],\, [1 \perp\!\!\!\perp 3|\emptyset],\, [1 \perp\!\!\!\perp 4|\emptyset],\, [2 \perp\!\!\!\perp 3|\emptyset],\, [2 \perp\!\!\!\perp 4|\emptyset],\, [3 \perp\!\!\!\perp 4|\emptyset], \\ [3 \perp\!\!\!\perp 4|12],\, [2 \perp\!\!\!\perp 4|13],\, [2 \perp\!\!\!\perp 3|14],\, [1 \perp\!\!\!\perp 4|23],\, [1 \perp\!\!\!\perp 3|24],\, [1 \perp\!\!\!\perp 2|34] \end{array} \right\rangle .$$

The semigraphoid ideal $I_{\mathbf{SG}}$ has 12 associated primes of codimension 15, such as

$$\left\langle \begin{array}{c} [1 \perp\!\!\!\perp 2|\emptyset],\, [1 \perp\!\!\!\perp 3|\emptyset],\, [1 \perp\!\!\!\perp 4|\emptyset],\, [3 \perp\!\!\!\perp 4|\emptyset],\, [1 \perp\!\!\!\perp 3|2],\, [1 \perp\!\!\!\perp 4|2],\, [3 \perp\!\!\!\perp 4|2],\, [1 \perp\!\!\!\perp 2|3], \\ [2 \perp\!\!\!\perp 4|3],\, [1 \perp\!\!\!\perp 2|4],\, [2 \perp\!\!\!\perp 3|4],\, [3 \perp\!\!\!\perp 4|12],\, [2 \perp\!\!\!\perp 4|13],\, [2 \perp\!\!\!\perp 3|14],\, [1 \perp\!\!\!\perp 2|34] \end{array} \right\rangle .$$

Next, $I_{\mathbf{SG}}$ has three associated primes of codimension 16. A representative is

$$\left\langle \begin{array}{c} [1\perp\!\!\!\perp 2|\emptyset], [1\perp\!\!\!\perp 3|\emptyset], [2\perp\!\!\!\perp 4|\emptyset], [3\perp\!\!\!\perp 4|\emptyset], [2\perp\!\!\!\perp 4|1], [3\perp\!\!\!\perp 4|1], [1\perp\!\!\!\perp 3|2], [3\perp\!\!\!\perp 4|2], \\ [1\perp\!\!\!\perp 2|3], [2\perp\!\!\!\perp 4|3], [1\perp\!\!\!\perp 2|4], [1\perp\!\!\!\perp 3|4], [3\perp\!\!\!\perp 4|12], [2\perp\!\!\!\perp 4|13], [1\perp\!\!\!\perp 3|24], [1\perp\!\!\!\perp 2|34] \end{array} \right\rangle .$$

Each of the 4316 non-submodular semigraphoids is a $\{0,1\}$-valued point not in $V(I_{\mathcal{A}})$ but in one of the 17 coordinate subspaces corresponding to these primes.

Finally, the last associated prime of $I_{\mathbf{SG}}$ is the toric ideal $I_{\mathcal{A}}$. This ideal has codimension 13 and degree 396. Its minimal generating set consists of 52 binomials. Besides the 24 quadrics (the axioms), the Markov basis of $\mathcal{A}$ contains four cubics

$$\begin{array}{l} [2\perp\!\!\!\perp 3|1] \cdot [3\perp\!\!\!\perp 4|2] \cdot [1\perp\!\!\!\perp 3|4] - [3\perp\!\!\!\perp 4|1] \cdot [1\perp\!\!\!\perp 3|2] \cdot [2\perp\!\!\!\perp 3|4], \\ [2\perp\!\!\!\perp 3|1] \cdot [2\perp\!\!\!\perp 4|3] \cdot [1\perp\!\!\!\perp 2|4] - [2\perp\!\!\!\perp 4|1] \cdot [1\perp\!\!\!\perp 2|3] \cdot [2\perp\!\!\!\perp 3|4], \\ [1\perp\!\!\!\perp 3|2] \cdot [1\perp\!\!\!\perp 4|3] \cdot [1\perp\!\!\!\perp 2|4] - [1\perp\!\!\!\perp 4|2] \cdot [1\perp\!\!\!\perp 2|3] \cdot [1\perp\!\!\!\perp 3|4], \\ [2\perp\!\!\!\perp 4|1] \cdot [3\perp\!\!\!\perp 4|2] \cdot [1\perp\!\!\!\perp 4|3] - [3\perp\!\!\!\perp 4|1] \cdot [1\perp\!\!\!\perp 4|2] \cdot [2\perp\!\!\!\perp 4|3], \end{array}$$

and 24 quartics such as

$$[1\perp\!\!\!\perp 2|\emptyset] \cdot [3\perp\!\!\!\perp 4|\emptyset] \cdot [2\perp\!\!\!\perp 4|13] \cdot [1\perp\!\!\!\perp 3|24] \;-\; [1\perp\!\!\!\perp 3|\emptyset] \cdot [2\perp\!\!\!\perp 4|\emptyset] \cdot [3\perp\!\!\!\perp 4|12] \cdot [1\perp\!\!\!\perp 2|34].$$

We now come to case $n = 5$. It will be a challenge for future commutative algebra software to compute a primary decomposition of the semigraphoid ideal $I_{\mathbf{SG}}$ for $n = 5$. At present we do not know even whether $I_{\mathbf{SG}}$ is radical. Let us therefore focus on the main component of this ideal, namely, the toric ideal $I_{\mathcal{A}}$. Here the main goal is to compute its minimal generators, that is, the Markov basis of $\mathcal{A}$. We attacked this problem using the software `4ti2` [39], and we now discuss the results.

First, we started a Markov basis computation for the toric ideal $I_{\mathcal{A}}$ using the function `markov` of `4ti2`, but this computation turned out to be non-trivial. In the hope that a counterexample would not involve all 80 variables, we set several variables to 0 and tried to compute the Markov basis of smaller ideals that are contained in $I_{\mathcal{A}}$. For the one-day computation that finally produced a counterexample, we set the first 18 formal symbols to zero and found the Markov basis move

$$\mathbf{g} \;:=\; \big(\,\alpha + 2 \cdot [2\perp\!\!\!\perp 3|45]\,\big) \;-\; \big(\,\beta + 2 \cdot [4\perp\!\!\!\perp 5|23]\,\big) \;\in\; \mathbb{N}^{80}, \qquad \text{where}$$

$$\alpha = [4\perp\!\!\!\perp 5|2] + [4\perp\!\!\!\perp 5|3] + [1\perp\!\!\!\perp 3|4] + [1\perp\!\!\!\perp 2|5] + [2\perp\!\!\!\perp 5|14] + [3\perp\!\!\!\perp 4|15] + [1\perp\!\!\!\perp 4|23] + [1\perp\!\!\!\perp 5|23],$$

$$\beta = [1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 4|3] + [2\perp\!\!\!\perp 3|4] + [2\perp\!\!\!\perp 3|5] + [3\perp\!\!\!\perp 4|12] + [2\perp\!\!\!\perp 5|13] + [1\perp\!\!\!\perp 2|45] + [1\perp\!\!\!\perp 3|45].$$

This lattice vector corresponds to a binomial $\mathbf{x}^{\mathbf{g}^+} - \mathbf{x}^{\mathbf{g}^-}$ which is in the toric ideal $I_{\mathcal{A}}$ and has the property that both of its monomials are not square-free. We then verified that $\mathbf{x}^{\mathbf{g}^+} - \mathbf{x}^{\mathbf{g}^-}$ is not only indispensable for the smaller ideal (with 18 variables set to zero) but also indispensable for $I_{\mathcal{A}}$. Recall (e.g. from [3]) that a binomial $\mathbf{x}^{\mathbf{g}^+} - \mathbf{x}^{\mathbf{g}^-}$ in the toric ideal $I_{\mathcal{A}}$ is called *indispensable* if

$$\{\mathbf{z} \in \mathbb{N}^{80} : \mathcal{A} \cdot \mathbf{z} = \mathcal{A} \cdot \mathbf{g}^+\} \quad = \quad \{\mathbf{g}^+, \mathbf{g}^-\}.$$

This means that the Markov move $\mathbf{g}$ corresponds to a 2-element fiber given by the right-hand side and consequently, $\mathbf{g}$ must belong to *every* Markov basis of $I_{\mathcal{A}}$. In order to

check this condition for the given move $\mathbf{g}$, we computed the minimal Hilbert basis (that is, the $\leq$-minimal integer solutions) of the cone

$$\{(\mathbf{z}, u) \in \mathbb{R}^{81} : \mathcal{A} \cdot \mathbf{z} - (\mathcal{A} \cdot \mathbf{g}^+) \cdot u = 0, (\mathbf{z}, u) \geq 0\}.$$

This was done using the function `hilbert` of `4ti2` which produced precisely the two expected elements $(\mathbf{g}^+, 1)$ and $(\mathbf{g}^-, 1)$ within a few seconds.

From the special Markov move $\mathbf{g} = (\alpha + 2 \cdot [2 \perp\!\!\!\perp 3|45]) - (\beta + 2 \cdot [4 \perp\!\!\!\perp 5|23])$, we then constructed the imset $\mathbf{b}$ presented in Section 4. We first checked that $\mathbf{b}$ was not a combinatorial imset by showing that $\mathcal{A}\mathbf{x} = \mathbf{b}$ has no solutions with non-negative integer coordinates. Using the functions `hilbert` and `rays` of the program `4ti2`, we computed the Hilbert basis and the extreme rays of the cone

$$\big\{(\mathbf{z}, u) \in \mathbb{R}^{81} : \mathcal{A} \cdot \mathbf{z} = \mathbf{b} \cdot u \text{ and } (\mathbf{z}, u) \geq 0\big\}.$$

Both computations quickly finished. They showed that this cone has dimension one and is generated by the single vector $(\alpha + \beta, 2)$. Consequently, the only non-negative real solution to $\mathcal{A} \cdot \mathbf{x} = \mathbf{b}$ is $(\alpha + \beta)/2$, which is not an integer solution.

7.6. Appendix: The 120 semigraphoid axioms

Here is the list of all 120 semigraphoid axiom for $n = 5$, grouped into triples according to which 3-face of the 5-cube they come from. The two types of brackets specify the non-submodular coarsest semigraphoid Γ which was discussed in Section 7.3.

$$
\begin{array}{lcl}
[3\perp\!\!\!\perp 5|12] + [\![3\perp\!\!\!\perp 4|125]\!] & = & [3\perp\!\!\!\perp 4|12] + [\![3\perp\!\!\!\perp 5|124]\!] \\
[4\perp\!\!\!\perp 5|12] + [\![3\perp\!\!\!\perp 4|125]\!] & = & [3\perp\!\!\!\perp 4|12] + [\![4\perp\!\!\!\perp 5|123]\!] \\
[4\perp\!\!\!\perp 5|12] + [\![3\perp\!\!\!\perp 5|124]\!] & = & [3\perp\!\!\!\perp 5|12] + [\![4\perp\!\!\!\perp 5|123]\!] \\
\\
[2\perp\!\!\!\perp 3|14] + [\![2\perp\!\!\!\perp 5|134]\!] & = & [2\perp\!\!\!\perp 5|14] + [\![2\perp\!\!\!\perp 3|145]\!] \\
[2\perp\!\!\!\perp 5|14] + [\![3\perp\!\!\!\perp 5|124]\!] & = & [3\perp\!\!\!\perp 5|14] + [\![2\perp\!\!\!\perp 5|134]\!] \\
[3\perp\!\!\!\perp 5|14] + [\![2\perp\!\!\!\perp 3|145]\!] & = & [2\perp\!\!\!\perp 3|14] + [\![3\perp\!\!\!\perp 5|124]\!] \\
\\
[1\perp\!\!\!\perp 5|23] + [\![1\perp\!\!\!\perp 4|235]\!] & = & [1\perp\!\!\!\perp 4|23] + [\![1\perp\!\!\!\perp 5|234]\!] \\
[4\perp\!\!\!\perp 5|23] + [\![1\perp\!\!\!\perp 4|235]\!] & = & [1\perp\!\!\!\perp 4|23] + [\![4\perp\!\!\!\perp 5|123]\!] \\
[4\perp\!\!\!\perp 5|23] + [\![1\perp\!\!\!\perp 5|234]\!] & = & [1\perp\!\!\!\perp 5|23] + [\![4\perp\!\!\!\perp 5|123]\!] \\
\\
[1\perp\!\!\!\perp 3|25] + [\![1\perp\!\!\!\perp 4|235]\!] & = & [1\perp\!\!\!\perp 4|25] + [\![1\perp\!\!\!\perp 3|245]\!] \\
[1\perp\!\!\!\perp 3|25] + [\![3\perp\!\!\!\perp 4|125]\!] & = & [3\perp\!\!\!\perp 4|25] + [\![1\perp\!\!\!\perp 3|245]\!] \\
[3\perp\!\!\!\perp 4|25] + [\![1\perp\!\!\!\perp 4|235]\!] & = & [1\perp\!\!\!\perp 4|25] + [\![3\perp\!\!\!\perp 4|125]\!] \\
\\
[1\perp\!\!\!\perp 2|35] + [\![1\perp\!\!\!\perp 4|235]\!] & = & [1\perp\!\!\!\perp 4|35] + [\![1\perp\!\!\!\perp 2|345]\!] \\
[1\perp\!\!\!\perp 2|35] + [\![2\perp\!\!\!\perp 4|135]\!] & = & [2\perp\!\!\!\perp 4|35] + [\![1\perp\!\!\!\perp 2|345]\!] \\
[1\perp\!\!\!\perp 4|35] + [\![2\perp\!\!\!\perp 4|135]\!] & = & [2\perp\!\!\!\perp 4|35] + [\![1\perp\!\!\!\perp 4|235]\!] \\
\\
[\![2\perp\!\!\!\perp 4|1]\!] + [3\perp\!\!\!\perp 4|12] & = & [\![3\perp\!\!\!\perp 4|1]\!] + [2\perp\!\!\!\perp 4|13] \\
[\![2\perp\!\!\!\perp 4|1]\!] + [2\perp\!\!\!\perp 3|14] & = & [\![2\perp\!\!\!\perp 3|1]\!] + [2\perp\!\!\!\perp 4|13] \\
[\![3\perp\!\!\!\perp 4|1]\!] + [2\perp\!\!\!\perp 3|14] & = & [\![2\perp\!\!\!\perp 3|1]\!] + [3\perp\!\!\!\perp 4|12] \\
\\
[\![2\perp\!\!\!\perp 4|1]\!] + [4\perp\!\!\!\perp 5|12] & = & [4\perp\!\!\!\perp 5|1] + [\![2\perp\!\!\!\perp 4|15]\!] \\
[2\perp\!\!\!\perp 5|1] + [\![2\perp\!\!\!\perp 4|15]\!] & = & [\![2\perp\!\!\!\perp 4|1]\!] + [2\perp\!\!\!\perp 5|14] \\
[4\perp\!\!\!\perp 5|1] + [2\perp\!\!\!\perp 5|14] & = & [2\perp\!\!\!\perp 5|1] + [4\perp\!\!\!\perp 5|12] \\
\\
[\![1\perp\!\!\!\perp 3|2]\!] + [1\perp\!\!\!\perp 4|23] & = & [\![1\perp\!\!\!\perp 4|2]\!] + [1\perp\!\!\!\perp 3|24] \\
[\![1\perp\!\!\!\perp 4|2]\!] + [3\perp\!\!\!\perp 4|12] & = & [\![3\perp\!\!\!\perp 4|2]\!] + [1\perp\!\!\!\perp 4|23] \\
[\![3\perp\!\!\!\perp 4|2]\!] + [1\perp\!\!\!\perp 3|24] & = & [\![1\perp\!\!\!\perp 3|2]\!] + [3\perp\!\!\!\perp 4|12] \\
\\
[\![1\perp\!\!\!\perp 4|2]\!] + [4\perp\!\!\!\perp 5|12] & = & [\![4\perp\!\!\!\perp 5|2]\!] + [1\perp\!\!\!\perp 4|25] \\
[\![1\perp\!\!\!\perp 4|2]\!] + [1\perp\!\!\!\perp 5|24] & = & [1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 4|25] \\
[1\perp\!\!\!\perp 5|2] + [4\perp\!\!\!\perp 5|12] & = & [\![4\perp\!\!\!\perp 5|2]\!] + [1\perp\!\!\!\perp 5|24] \\
\\
[\![1\perp\!\!\!\perp 4|3]\!] + [2\perp\!\!\!\perp 4|13] & = & [\![2\perp\!\!\!\perp 4|3]\!] + [1\perp\!\!\!\perp 4|23] \\
[\![1\perp\!\!\!\perp 4|3]\!] + [1\perp\!\!\!\perp 2|34] & = & [\![1\perp\!\!\!\perp 2|3]\!] + [1\perp\!\!\!\perp 4|23] \\
[\![2\perp\!\!\!\perp 4|3]\!] + [1\perp\!\!\!\perp 2|34] & = & [\![1\perp\!\!\!\perp 2|3]\!] + [2\perp\!\!\!\perp 4|13] \\
\\
[\![1\perp\!\!\!\perp 4|3]\!] + [4\perp\!\!\!\perp 5|13] & = & [\![4\perp\!\!\!\perp 5|3]\!] + [1\perp\!\!\!\perp 4|35] \\
[\![1\perp\!\!\!\perp 4|3]\!] + [1\perp\!\!\!\perp 5|34] & = & [1\perp\!\!\!\perp 5|3] + [1\perp\!\!\!\perp 4|35] \\
[\![4\perp\!\!\!\perp 5|3]\!] + [1\perp\!\!\!\perp 5|34] & = & [1\perp\!\!\!\perp 5|3] + [4\perp\!\!\!\perp 5|13] \\
\\
[\![1\perp\!\!\!\perp 2|4]\!] + [2\perp\!\!\!\perp 3|14] & = & [\![2\perp\!\!\!\perp 3|4]\!] + [1\perp\!\!\!\perp 2|34] \\
[\![1\perp\!\!\!\perp 2|4]\!] + [1\perp\!\!\!\perp 3|24] & = & [\![1\perp\!\!\!\perp 3|4]\!] + [1\perp\!\!\!\perp 2|34] \\
[\![1\perp\!\!\!\perp 3|4]\!] + [2\perp\!\!\!\perp 3|14] & = & [\![2\perp\!\!\!\perp 3|4]\!] + [1\perp\!\!\!\perp 3|24] \\
\\
[\![1\perp\!\!\!\perp 3|4]\!] + [1\perp\!\!\!\perp 5|34] & = & [1\perp\!\!\!\perp 5|4] + [1\perp\!\!\!\perp 3|45] \\
[\![3\perp\!\!\!\perp 5|4]\!] + [1\perp\!\!\!\perp 5|34] & = & [1\perp\!\!\!\perp 5|4] + [3\perp\!\!\!\perp 5|14] \\
[\![3\perp\!\!\!\perp 5|4]\!] + [1\perp\!\!\!\perp 3|45] & = & [\![1\perp\!\!\!\perp 3|4]\!] + [3\perp\!\!\!\perp 5|14] \\
\\
[1\perp\!\!\!\perp 2|5] + [\![2\perp\!\!\!\perp 3|15]\!] & = & [\![2\perp\!\!\!\perp 3|5]\!] + [1\perp\!\!\!\perp 2|35] \\
[1\perp\!\!\!\perp 2|5] + [1\perp\!\!\!\perp 3|25] & = & [1\perp\!\!\!\perp 3|5] + [1\perp\!\!\!\perp 2|35] \\
[1\perp\!\!\!\perp 3|5] + [\![2\perp\!\!\!\perp 3|15]\!] & = & [\![2\perp\!\!\!\perp 3|5]\!] + [1\perp\!\!\!\perp 3|25] \\
\\
[1\perp\!\!\!\perp 3|5] + [\![3\perp\!\!\!\perp 4|15]\!] & = & [\![3\perp\!\!\!\perp 4|5]\!] + [1\perp\!\!\!\perp 3|45] \\
[1\perp\!\!\!\perp 3|5] + [1\perp\!\!\!\perp 4|35] & = & [1\perp\!\!\!\perp 4|5] + [1\perp\!\!\!\perp 3|45] \\
[\![3\perp\!\!\!\perp 4|5]\!] + [1\perp\!\!\!\perp 4|35] & = & [1\perp\!\!\!\perp 4|5] + [\![3\perp\!\!\!\perp 4|15]\!] \\
\\
[\![1\perp\!\!\!\perp 2|]\!] + [\![2\perp\!\!\!\perp 3|1]\!] & = & [\![2\perp\!\!\!\perp 3|]\!] + [\![1\perp\!\!\!\perp 2|3]\!] \\
[\![1\perp\!\!\!\perp 3|]\!] + [\![1\perp\!\!\!\perp 2|3]\!] & = & [\![1\perp\!\!\!\perp 2|]\!] + [\![1\perp\!\!\!\perp 3|2]\!] \\
[\![2\perp\!\!\!\perp 3|]\!] + [\![1\perp\!\!\!\perp 3|2]\!] & = & [\![1\perp\!\!\!\perp 3|]\!] + [\![2\perp\!\!\!\perp 3|1]\!] \\
\\
[\![1\perp\!\!\!\perp 2|]\!] + [2\perp\!\!\!\perp 5|1] & = & [\![2\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 2|5] \\
[\![1\perp\!\!\!\perp 2|]\!] + [1\perp\!\!\!\perp 5|2] & = & [\![1\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 2|5] \\
[\![1\perp\!\!\!\perp 5|]\!] + [2\perp\!\!\!\perp 5|1] & = & [\![2\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 5|2] \\
\\
[\![1\perp\!\!\!\perp 3|]\!] + [3\perp\!\!\!\perp 5|1] & = & [\![3\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 3|5] \\
[\![1\perp\!\!\!\perp 5|]\!] + [3\perp\!\!\!\perp 5|1] & = & [\![3\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 5|3] \\
[\![1\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 3|5] & = & [\![1\perp\!\!\!\perp 3|]\!] + [1\perp\!\!\!\perp 5|3] \\
\\
[\![2\perp\!\!\!\perp 4|]\!] + [\![2\perp\!\!\!\perp 3|4]\!] & = & [\![2\perp\!\!\!\perp 3|]\!] + [\![2\perp\!\!\!\perp 4|3]\!] \\
[\![3\perp\!\!\!\perp 4|]\!] + [\![2\perp\!\!\!\perp 4|3]\!] & = & [\![2\perp\!\!\!\perp 4|]\!] + [\![3\perp\!\!\!\perp 4|2]\!] \\
[\![3\perp\!\!\!\perp 4|]\!] + [\![2\perp\!\!\!\perp 3|4]\!] & = & [\![2\perp\!\!\!\perp 3|]\!] + [\![3\perp\!\!\!\perp 4|2]\!] \\
\\
[\![2\perp\!\!\!\perp 4|]\!] + [\![4\perp\!\!\!\perp 5|2]\!] & = & [\![4\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 4|5]\!] \\
[\![2\perp\!\!\!\perp 5|]\!] + [\![4\perp\!\!\!\perp 5|2]\!] & = & [\![4\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 5|4]\!] \\
[\![2\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 4|5]\!] & = & [\![2\perp\!\!\!\perp 4|]\!] + [\![2\perp\!\!\!\perp 5|4]\!]
\end{array}
\qquad
\begin{array}{lcl}
[\![2\perp\!\!\!\perp 5|13] + [\![2\perp\!\!\!\perp 4|135]\!] & = & [\![2\perp\!\!\!\perp 4|13] + [\![2\perp\!\!\!\perp 5|134]\!] \\
[4\perp\!\!\!\perp 5|13] + [\![2\perp\!\!\!\perp 4|135]\!] & = & [2\perp\!\!\!\perp 4|13] + [\![4\perp\!\!\!\perp 5|123]\!] \\
[4\perp\!\!\!\perp 5|13] + [\![2\perp\!\!\!\perp 5|134]\!] & = & [2\perp\!\!\!\perp 5|13] + [\![4\perp\!\!\!\perp 5|123]\!] \\
\\
[\![2\perp\!\!\!\perp 4|15]\!] + [\![2\perp\!\!\!\perp 3|145]\!] & = & [\![2\perp\!\!\!\perp 3|15]\!] + [\![2\perp\!\!\!\perp 4|135]\!] \\
[\![3\perp\!\!\!\perp 4|15]\!] + [\![2\perp\!\!\!\perp 3|145]\!] & = & [\![2\perp\!\!\!\perp 3|15]\!] + [\![3\perp\!\!\!\perp 4|125]\!] \\
[\![3\perp\!\!\!\perp 4|15]\!] + [\![2\perp\!\!\!\perp 4|135]\!] & = & [\![2\perp\!\!\!\perp 4|15]\!] + [\![3\perp\!\!\!\perp 4|125]\!] \\
\\
[1\perp\!\!\!\perp 3|24] + [\![3\perp\!\!\!\perp 5|124]\!] & = & [3\perp\!\!\!\perp 5|24] + [\![1\perp\!\!\!\perp 3|245]\!] \\
[1\perp\!\!\!\perp 5|24] + [\![1\perp\!\!\!\perp 3|245]\!] & = & [1\perp\!\!\!\perp 3|24] + [\![1\perp\!\!\!\perp 5|234]\!] \\
[1\perp\!\!\!\perp 5|24] + [\![3\perp\!\!\!\perp 5|124]\!] & = & [3\perp\!\!\!\perp 5|24] + [\![1\perp\!\!\!\perp 5|234]\!] \\
\\
[1\perp\!\!\!\perp 2|34] + [\![2\perp\!\!\!\perp 5|134]\!] & = & [2\perp\!\!\!\perp 5|34] + [\![1\perp\!\!\!\perp 2|345]\!] \\
[1\perp\!\!\!\perp 5|34] + [\![1\perp\!\!\!\perp 2|345]\!] & = & [1\perp\!\!\!\perp 2|34] + [\![1\perp\!\!\!\perp 5|234]\!] \\
[1\perp\!\!\!\perp 5|34] + [\![2\perp\!\!\!\perp 5|134]\!] & = & [2\perp\!\!\!\perp 5|34] + [\![1\perp\!\!\!\perp 5|234]\!] \\
\\
[1\perp\!\!\!\perp 2|45] + [\![1\perp\!\!\!\perp 3|245]\!] & = & [1\perp\!\!\!\perp 3|45] + [\![1\perp\!\!\!\perp 2|345]\!] \\
[1\perp\!\!\!\perp 3|45] + [\![2\perp\!\!\!\perp 3|145]\!] & = & [2\perp\!\!\!\perp 3|45] + [\![1\perp\!\!\!\perp 3|245]\!] \\
[2\perp\!\!\!\perp 3|45] + [\![1\perp\!\!\!\perp 2|345]\!] & = & [1\perp\!\!\!\perp 2|45] + [\![2\perp\!\!\!\perp 3|145]\!] \\
\\
[\![2\perp\!\!\!\perp 3|1]\!] + [3\perp\!\!\!\perp 5|12] & = & [3\perp\!\!\!\perp 5|1] + [\![2\perp\!\!\!\perp 3|15]\!] \\
[\![2\perp\!\!\!\perp 3|1]\!] + [2\perp\!\!\!\perp 5|13] & = & [2\perp\!\!\!\perp 5|1] + [\![2\perp\!\!\!\perp 3|15]\!] \\
[3\perp\!\!\!\perp 5|1] + [2\perp\!\!\!\perp 5|13] & = & [2\perp\!\!\!\perp 5|1] + [3\perp\!\!\!\perp 5|12] \\
\\
[3\perp\!\!\!\perp 5|1] + [4\perp\!\!\!\perp 5|13] & = & [4\perp\!\!\!\perp 5|1] + [3\perp\!\!\!\perp 5|14] \\
[3\perp\!\!\!\perp 5|1] + [\![3\perp\!\!\!\perp 4|15]\!] & = & [\![3\perp\!\!\!\perp 4|1]\!] + [3\perp\!\!\!\perp 5|14] \\
[4\perp\!\!\!\perp 5|1] + [\![3\perp\!\!\!\perp 4|15]\!] & = & [\![3\perp\!\!\!\perp 4|1]\!] + [4\perp\!\!\!\perp 5|13] \\
\\
[1\perp\!\!\!\perp 5|2] + [1\perp\!\!\!\perp 3|25] & = & [\![1\perp\!\!\!\perp 3|2]\!] + [1\perp\!\!\!\perp 5|23] \\
[\![3\perp\!\!\!\perp 5|2]\!] + [1\perp\!\!\!\perp 5|23] & = & [1\perp\!\!\!\perp 5|2] + [3\perp\!\!\!\perp 5|12] \\
[\![3\perp\!\!\!\perp 5|2]\!] + [1\perp\!\!\!\perp 3|25] & = & [\![1\perp\!\!\!\perp 3|2]\!] + [3\perp\!\!\!\perp 5|12] \\
\\
[\![3\perp\!\!\!\perp 5|2]\!] + [4\perp\!\!\!\perp 5|23] & = & [\![4\perp\!\!\!\perp 5|2]\!] + [3\perp\!\!\!\perp 5|24] \\
[\![3\perp\!\!\!\perp 5|2]\!] + [3\perp\!\!\!\perp 4|25] & = & [\![3\perp\!\!\!\perp 4|2]\!] + [3\perp\!\!\!\perp 5|24] \\
[\![4\perp\!\!\!\perp 5|2]\!] + [3\perp\!\!\!\perp 4|25] & = & [\![3\perp\!\!\!\perp 4|2]\!] + [4\perp\!\!\!\perp 5|23] \\
\\
[\![1\perp\!\!\!\perp 2|3]\!] + [2\perp\!\!\!\perp 5|13] & = & [\![2\perp\!\!\!\perp 5|3]\!] + [1\perp\!\!\!\perp 2|35] \\
[1\perp\!\!\!\perp 5|3] + [1\perp\!\!\!\perp 2|35] & = & [\![1\perp\!\!\!\perp 2|3]\!] + [1\perp\!\!\!\perp 5|23] \\
[\![2\perp\!\!\!\perp 5|3]\!] + [1\perp\!\!\!\perp 5|23] & = & [1\perp\!\!\!\perp 5|3] + [2\perp\!\!\!\perp 5|13] \\
\\
[\![2\perp\!\!\!\perp 4|3]\!] + [4\perp\!\!\!\perp 5|23] & = & [\![4\perp\!\!\!\perp 5|3]\!] + [2\perp\!\!\!\perp 4|35] \\
[\![2\perp\!\!\!\perp 5|3]\!] + [2\perp\!\!\!\perp 4|35] & = & [\![2\perp\!\!\!\perp 4|3]\!] + [2\perp\!\!\!\perp 5|34] \\
[\![4\perp\!\!\!\perp 5|3]\!] + [2\perp\!\!\!\perp 5|34] & = & [\![2\perp\!\!\!\perp 5|3]\!] + [4\perp\!\!\!\perp 5|23] \\
\\
[\![1\perp\!\!\!\perp 2|4]\!] + [2\perp\!\!\!\perp 5|14] & = & [\![2\perp\!\!\!\perp 5|4]\!] + [1\perp\!\!\!\perp 2|45] \\
[\![1\perp\!\!\!\perp 2|4]\!] + [1\perp\!\!\!\perp 5|24] & = & [1\perp\!\!\!\perp 5|4] + [1\perp\!\!\!\perp 2|45] \\
[1\perp\!\!\!\perp 5|4] + [2\perp\!\!\!\perp 5|14] & = & [\![2\perp\!\!\!\perp 5|4]\!] + [1\perp\!\!\!\perp 5|24] \\
\\
[\![2\perp\!\!\!\perp 3|4]\!] + [3\perp\!\!\!\perp 5|24] & = & [\![3\perp\!\!\!\perp 5|4]\!] + [2\perp\!\!\!\perp 3|45] \\
[\![2\perp\!\!\!\perp 3|4]\!] + [2\perp\!\!\!\perp 5|34] & = & [\![2\perp\!\!\!\perp 5|4]\!] + [2\perp\!\!\!\perp 3|45] \\
[\![2\perp\!\!\!\perp 5|4]\!] + [3\perp\!\!\!\perp 5|24] & = & [\![3\perp\!\!\!\perp 5|4]\!] + [2\perp\!\!\!\perp 5|34] \\
\\
[\![1\perp\!\!\!\perp 2|5]\!] + [\![2\perp\!\!\!\perp 4|15]\!] & = & [\![2\perp\!\!\!\perp 4|5]\!] + [1\perp\!\!\!\perp 2|45] \\
[1\perp\!\!\!\perp 2|5] + [1\perp\!\!\!\perp 4|25] & = & [1\perp\!\!\!\perp 4|5] + [1\perp\!\!\!\perp 2|45] \\
[1\perp\!\!\!\perp 4|5] + [\![2\perp\!\!\!\perp 4|15]\!] & = & [\![2\perp\!\!\!\perp 4|5]\!] + [1\perp\!\!\!\perp 4|25] \\
\\
[\![2\perp\!\!\!\perp 3|5]\!] + [2\perp\!\!\!\perp 4|35] & = & [\![2\perp\!\!\!\perp 4|5]\!] + [2\perp\!\!\!\perp 3|45] \\
[\![2\perp\!\!\!\perp 4|5]\!] + [3\perp\!\!\!\perp 4|25] & = & [\![3\perp\!\!\!\perp 4|5]\!] + [2\perp\!\!\!\perp 4|35] \\
[\![3\perp\!\!\!\perp 4|5]\!] + [2\perp\!\!\!\perp 3|45] & = & [\![2\perp\!\!\!\perp 3|5]\!] + [3\perp\!\!\!\perp 4|25] \\
\\
[\![1\perp\!\!\!\perp 2|]\!] + [\![2\perp\!\!\!\perp 4|1]\!] & = & [\![2\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 2|4]\!] \\
[\![1\perp\!\!\!\perp 2|]\!] + [\![1\perp\!\!\!\perp 4|2]\!] & = & [\![1\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 2|4]\!] \\
[\![1\perp\!\!\!\perp 4|]\!] + [\![2\perp\!\!\!\perp 4|1]\!] & = & [\![2\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 4|2]\!] \\
\\
[\![1\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 3|4]\!] & = & [\![1\perp\!\!\!\perp 3|]\!] + [\![1\perp\!\!\!\perp 4|3]\!] \\
[\![3\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 4|3]\!] & = & [\![1\perp\!\!\!\perp 4|]\!] + [\![3\perp\!\!\!\perp 4|1]\!] \\
[\![3\perp\!\!\!\perp 4|]\!] + [\![1\perp\!\!\!\perp 3|4]\!] & = & [\![1\perp\!\!\!\perp 3|]\!] + [\![3\perp\!\!\!\perp 4|1]\!] \\
\\
[\![1\perp\!\!\!\perp 4|]\!] + [1\perp\!\!\!\perp 5|4] & = & [\![1\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 4|5] \\
[\![4\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 5|4] & = & [\![1\perp\!\!\!\perp 5|]\!] + [4\perp\!\!\!\perp 5|1] \\
[\![4\perp\!\!\!\perp 5|]\!] + [1\perp\!\!\!\perp 4|5] & = & [\![1\perp\!\!\!\perp 4|]\!] + [4\perp\!\!\!\perp 5|1] \\
\\
[\![2\perp\!\!\!\perp 3|]\!] + [\![2\perp\!\!\!\perp 5|3]\!] & = & [\![2\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 3|5]\!] \\
[\![2\perp\!\!\!\perp 5|]\!] + [\![3\perp\!\!\!\perp 5|2]\!] & = & [\![3\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 5|3]\!] \\
[\![3\perp\!\!\!\perp 5|]\!] + [\![2\perp\!\!\!\perp 3|5]\!] & = & [\![2\perp\!\!\!\perp 3|]\!] + [\![3\perp\!\!\!\perp 5|2]\!] \\
\\
[\![3\perp\!\!\!\perp 4|]\!] + [\![4\perp\!\!\!\perp 5|3]\!] & = & [\![4\perp\!\!\!\perp 5|]\!] + [\![3\perp\!\!\!\perp 4|5]\!] \\
[\![3\perp\!\!\!\perp 4|]\!] + [\![3\perp\!\!\!\perp 5|4]\!] & = & [\![3\perp\!\!\!\perp 5|]\!] + [\![3\perp\!\!\!\perp 4|5]\!] \\
[\![3\perp\!\!\!\perp 5|]\!] + [\![4\perp\!\!\!\perp 5|3]\!] & = & [\![4\perp\!\!\!\perp 5|]\!] + [\![3\perp\!\!\!\perp 5|4]\!]
\end{array}
$$

Table 7.1.: List of all 120 semigraphoid axiom for $n = 5$

Chapter 8.

Computing distributive lattices and counting linear extensions

In Chapter 6 we have introduced a hierarchy of rank tests, which range from pre-convex to graphical. Convex rank tests are applied to data vectors $u \in \mathbb{R}^n$, or permutations $\pi \in S_n$, and determine their cones in a fan $\mathcal{F}$ which coarsens the S_n-fan. The significance of a data vector in such a test is measured by a certain p-value, whose precise derivation is described in [56]. Computation of that p-value rests on the ability to compute the quantity $|\,\tau^{-1}\big(\tau(\pi)\big)\,|$, which is the number of permutations in the maximal cone of $\mathcal{F}$ corresponding to π. Recall that the cones of a convex rank test are indexed by posets $P_1, P_2, \ldots, P_k$ on $[n]$, and the computations amount to finding the cardinality of the set $\mathcal{L}(P_i)$ of linear extensions of P_i.

The following Algorithms 8.1 and 8.2 are implemented in R [68] and Python [85]. The source code and manuals are available for download at

`http://bio.math.berkeley.edu/ranktests/`.

8.1. On counting linear extensions

The problem of computing linear extensions of general posets is #P-complete [10], so the task is an intractable problem when n grows large. However, for special classes of posets, and for moderate values of n, the situation is not so bad. For example, in the up-down analysis of Willbrand *et al.* (see Example 6.3.4), we need to count all permutations with a fixed descent set, a task for which an explicit determinantal formula appears in Stanley [77, page 69]. We refer to [11] for a detailed study of the combinatorics of these *up-down numbers*.

Likewise, there is an efficient (and easy-to-implement) method for the computing quantities $|\,\tau^{-1}\big(\tau(\pi)\big)\,|$ for any graphical graphical tubing test $\tau^*_{\mathcal{K}(G)}$, as defined in Section 5. Indeed, here the fan $\mathcal{F}$ is unimodular, and hence the posets P_i are all trees. The special trees arising from a graph G in this manner are known as *G-trees* [65, 18]. The G-tree of a permutation π is a representation of the poset P_i as a tree $T = \tau^*_{\mathcal{K}(G)}(\pi)$ with the minimum value as the root and maximal values as the leaves. Suppose the root of the tree T has k children, each of which is a root of a subtree T^i for $i = 1, \ldots, k$.

Writing $|T^i|$ for the number of nodes in T^i, we have

$$|\,\tau^{-1}(T)\,| \quad = \quad \binom{\sum_{i=1}^{k}|T^i|}{|T^1|,\ldots,|T^k|}\left(\prod_{i=1}^{k}|\tau^{-1}(T^i)|\right).$$

This recursive formula translates into an efficient iterative algorithm. The implementation of this algorithm, when G is the n-cycle, is the workhorse behind the computations in [56]. For a graph G, let nbhd(i) be the set of vertices j such that there is an edge (i,j) in G.

(Permutation Counting)
Input: A data point u as a descent permutation δ and a graph G.
Output: The number of permutations with the same signature as δ, $|\,\tau^{-1}\tau(\pi(u))\,|$.

Initialize:
An indexed set of largest enclosing sets $LE_1 = \cdots = LE_n = \emptyset$, and counter $c = 1$
for δ_i in δ:
  Initialize ℓ an empty list of enclosed tree lengths
  $LE_{\delta_i} = \{\delta_i\}$
  for j in nbhd(δ_i):
    if $LE_j \neq \emptyset$ and $j \notin LE_{\delta_i}$:
      $LE_{\delta_i} = LE_{\delta_i} \sqcup LE_j$
      append $|LE_j|$ to ℓ
  $c = c \cdot \binom{\sum_i(\ell_i)}{\ell}$
  for j in LE_{δ_i}:
    $LE_j = LE_{\delta_i}$
Return the permutation count c

The algorithm is implemented in R and was used to successfully rank all gene expressions measured in experiments concerning the development of the vertebra (see [24]).

8.2. Computing the distributive lattice of order ideals

In the remainder of this chapter we discuss our method for performing these computations for an arbitrary convex rank test. The test is specified (implicitly or explicitly) by a collection of posets $P_1,\ldots,P_k$ on $[n]$. From the given permutation, we identify the unique poset P_i of which that permutation is a linear extension, and we construct the *distributive lattice* $L(P_i)$ whose elements are the order ideals of P_i. Recall that an *order ideal* of P_i is a subset O of $[n]$ such that if $l \in O$ and $(k,l) \in P_i$ then $k \in O$. The set of all order ideals is a distributive lattice with meet and join operations given by set intersection $O \cap O'$ and set union $O \cup O'$.

The distributive lattice $L(P_i)$ is a sublattice of the Boolean lattice $2^{[n]}$, whose nodes are the 2^n subsets of $[n] = \{1, 2, \ldots, n\}$, and we represent $L(P_i)$ by its nodes and edges (cover relations) in $2^{[n]}$. We write each edge in $2^{[n]}$ as a pair (K, l) where $K \subset [n]$ and $l \in [n] \backslash K$. The edge in the Boolean lattice $2^{[n]}$ represented by the pair (K, l) is the cover relation $K \subset K \cup \{l\}$.

Permutations in S_n are in natural bijection with maximal chains in the Boolean lattice $2^{[n]}$. For example, the descent permutation $\delta = (4|2|1|3)$ corresponds to the maximal chain $\big(\emptyset, \{4\}, \{2,4\}, \{1,2,4\}, \{1,2,3,4\}\big)$ in the Boolean lattice $2^{[4]}$. If the poset P_i is the linear order δ then $L(P_i)$ is the subgraph of $2^{[4]}$ consisting of the five nodes in the chain and the four edges $(\emptyset, 4)$, $(\{4\}, 2)$, $(\{2,4\}, 1)$ and $(\{1,2,4\}, 3)$ which connect them. The maximal chains in $2^{[n]}$ that lie in the sublattice $L(P_i)$ are precisely the permutations that are linear extensions of P_i. Therefore our task is to construct $L(P_i)$ and then count its maximal chains.

Remark 8.2.1. *The linear extensions of the poset P_i are in bijection with the maximal chains in the distributive lattice $L(P_i)$. See [77, Section 3.5] for further information on this bijection.*

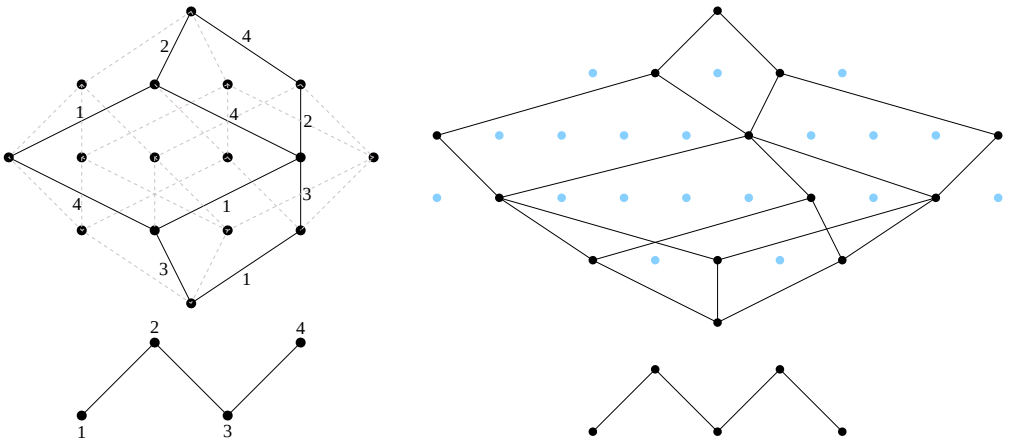


Figure 8.1.: The distributive lattice of a fence of length 4 and 5. In the first case all unused edges are indicated in light grey. In the second all unused vertices are depicted in lighter color. The total number of paths is 5, respectively 16

Example 8.2.2. *The figure 8.1 depicts distribute lattices whose contents are the linear extensions of the fence of length 4 resp. 5.*

In general, $L(P_i)$ is the graph whose nodes are those subsets of $[n]$ which are order ideals in P_i, and the edges are (K, l) where both K and $K \cup \{l\}$ are order ideals in P_i. Our strategy in computing the graph which represents $L(P_i)$ is as follows. We start with a given permutation δ which lies in the class indexed by P_i. That permutation determines a maximal chain in $2^{[n]}$ which must lie in $L(P_i)$. We then compute a certain closure of that subgraph in $2^{[n]}$ with respect to the semigraphoid $\mathcal{M}$ under consideration.

This is precisely what is done in Algorithm 8.2 below. Knowledge of the distributive lattice $L(P_i)$ solves our problem since the number of maximal chains of $L(P_i)$ can be read easily from the representation of $L(P_i)$ in terms of nodes and edges.

(Building the Distributive Lattice)
Input: A data point as a descent permutation δ and a semigraphoid $\mathcal{M}$.
Output: A distributive lattice $L(P_i)$ representing the class of δ in the convex rank test $\mathcal{M}$.

Initialize:
A set of confirmed lattice nodes, $\mathbb{H} = \big\{\emptyset, \{\delta_1\}, \{\delta_1, \delta_2\}, \ldots, \{\delta_1, \ldots, \delta_n\}\big\}$
A set of checked lattice edges, $E = \big\{(\{\delta_1, \ldots, \delta_{n-1}\}, \delta_n)\big\}$,
where each pair has the form (history, next position).
A stack of edges waiting to be checked:
$W = \big[(\emptyset, \delta_1), (\{\delta_1\}, \delta_2), (\{\delta_1, \delta_2\}, \delta_3), \ldots, (\{\delta_1, \ldots, \delta_{n-2}\}, \delta_{n-1})\big]$

While $W \neq \emptyset$:
  Pop (H, i) from the stack W
  Add (H, i) to E
  for j such that $(H \cup \{i\}, j) \in E$:
    if $i \perp\!\!\!\perp j | H \in \mathcal{M}$:
      Add (H, j) to E
      if $H \cup \{j\} \notin \mathbb{H}$:
        Add $H \cup \{j\}$ to $\mathbb{H}$
        Push $(H \cup \{j\}, i)$ onto W
Return the distributive lattice $L(P_i) = \big(\mathbb{H}, E\big)$

Our program for performing rank tests implements Algorithm 8.2. It accepts a permutation δ and a rank test τ, which may be specified either

- by a list of posets $P_1, \ldots, P_k$ (pre-convex),
- or by a semigraphoid $\mathcal{M}$ (convex rank test),
- or by a submodular function $w : 2^{[n]} \to \mathbb{R}$,
- or by a collection $\mathcal{K}$ of subsets of $[n]$ (MSS),
- or by a graph G on $[n]$ (graphical test)
- or by data vectors $u^{(1)}, \ldots, u^{(n)} \in \mathbb{R}^s$ (partial order ranking).

Once the programm has computed the distributive lattice of order ideals the following data is easy to compute and corresponding algorithms are implemented:

- the number $|\mathcal{L}(P_i)|$ of linear extensions, where the poset P_i represents the equivalence class of S_n specified by the data π,
- the number of vertices and edges in the distributive lattice $L(P_i)$,
- a truly uniform sample of all linear extensions of P_i,
- the rank frequencies of the different labels $l_i \in [n]$, i. e. the average position of each label in all linear extensions of P_i,
- a representation of the distributive lattice $L(P_i)$, in a format that can be read by the `maple` package `posets` [78],
- a test of modularity for the function w which gives a counterexamples if the function is neither sub- nor supermodular.

Our software for Algorithms 8.1 and 8.2 and, more generally, for applying convex rank tests τ to data vectors $u \in \mathbb{R}^n$ is available at `http://bio.math.berkeley.edu/ranktests/`.

8.3. Benchmarks

The only other implemented algorithm we found is available at

`http://webhome.cs.uvic.ca/~ruskey/Publications/ExtensionFast/ExtensionFast.html`

and described in [67].

We compared both algorithms by using the even fence posets (see Table 8.1). The number of linear extensions are well-known to be the coefficients in the expansion of $\tan(x)*x$ [83] and therefore easy to verify.

In closing let us give a concrete illustration of our current ability to count linear extensions. We computed the number of linear extensions of the Boolean poset $P = 2^{[5]}$

#elements	#extensions	lcell	genle	comment
12	4245504	0.2 s	<0.1 s	
14	313155584	0.2 s	0.4 s	
16	30460116992	0.3 s	23.2 s	wrong result by genle
18	3777576173568	0.5 s	>30min	
20	581777702256640	1.2 s		
22	108932957168730112	2.7 s		
24	24370173276164456448	14.5 s		

Table 8.1.: Computing the number of all linear extensions for fences on an Intel Core2 Duo P8600, 2.4 GHz, 2 GB RAM.

consisting of all subsets of $\{1, 2, 3, 4, 5\}$. Our program ran in less than one second on a laptop and found that

$$|L(2^{[5]})| \quad = \quad 14\,807\,804\,035\,657\,359\,360.$$

This computation was inspired by work in population genetics of Daniel Weinreich [88] who reports the analogous calculation for $P = 2^{[4]}$. The computed number $|L(2^{[5]})|$ was previously unknown and extends the sequence A046873 of the On-Line Encyclopedia of Integer Sequences (see [82]). We also computed

$$|L(2^{[6]})| = 141\,377\,911\,697\,227\,887\,117\,195\,970\,316\,200\,795\,630\,205\,476\,957\,716\,480$$

on a compute server in less than 16 hours.

Bibliography

[1] W. Adams and P. Loustaunau. *An introduction to Gröbner bases*. (Graduate studies in mathematics) AMS, 2003.

[2] D. D. Anderson and S. Valdes-Leon. Factorization in commutative rings with zero divisors. *Rocky Moutain Journal of Mathematics*, 26(2):439–480, 1996.

[3] S. Aoki, A. Takemura, and R. Yoshida. Indispensable monomials of toric ideals and markov bases. *Journal of Symbolic Computation*, 43 (6-7):490–507, 2008.

[4] D. Babic. Spear, `http://www.domagoj-babic.com`.

[5] L. Billera, I. Gelfand, and B. Sturmfels. Duality and minors secondary polyhedra. *Journal of Combinatorial Theory Ser. B*, 57:258–268, 1993.

[6] J. Bokowski and B. Sturmfels. Polytopal and non-polytopal spheres—an algorithmic approach. *Israel Journal of Mathematics*, 57:257–271, 1987.

[7] W. Bosma, J. Cannon, and C. Playoust. The MAGMA algebra system I: the user language. *J. Symb. Comput.*, 24 (3-4):235–265, 1997.

[8] M. Brickenstein. *Boolean Gröbner bases - "Theory, Algorithms and Applications"*. Logos Verlag Berlin GmbH, 2010.

[9] M. Brickenstein, A. Dreyer, G.-M. Greuel, M. Wedler, and O. Wienand. New developments in the theory of Gröbner bases and applications to formal verification. *Journal of Pure and Applied Algebra*, 213:1612–1635, 2009.

[10] G. Brightwell and P. Winkler. Counting linear extension. *Order*, 8:225–242, 1991.

[11] F. Brown, T. Fink, and K. Willbrand. On arithmetic and asymptotic properties of up-down numbers. *Discrete Mathematics*, 307:1722–1736, 2007.

[12] K. Brown. *Buildings*. Springer, New York, 1989.

[13] R. Brüggemann and G. Patil. *Ranking and Prioritization for Multi-Indicator Systems - Introduction to Partial Order Applications*. Springer-Verlag, 2011.

[14] W. Bruns and R. Koch. Computing the integral closure of an affine semigroup. effective methods in algebraic and analytic geometry. *Univ. Iagel. Acta Math.*, 39:59–70, 2001.

[15] R. Bruttomesso, A. Cimatti, A. Franzén, A. Griggio, Z. Hanna, A. Nadel, and R. S. A Palti. A lazy and layered smt($\{BV\}$) solver for hard industrial verification problems. *CAV'07 Proceedings of the 19th international conference on Computer aided verification*, pages 547–560, 2007.

[16] B. Buchberger. *An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal (in German).* PhD thesis, Univ. Innsbruck, Dept. of Math., Innsbruck, Austria, 1965.

[17] L. Carlitz. Functions and polynomials (mod p^n). *Acta Arith*, 9:67–78, 1964.

[18] M. Carr and S. Devadoss. Coxeter complexes and graph associahedra. *Topology and its Applications*, 153:2155–2168, 2006.

[19] Center for Mathematical and Computational Modelling $(CM)^2$. Further information available at `http://cmcm.uni-kl.de/en/`.

[20] W. Cook and L. Seiford. The geometry of rank-order tests. *The American Statistician*, 37:307–311, 1983.

[21] A. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society B*, 41:1–31, 1979.

[22] L. M. de Moura and N. Bjoerner. Efficient e-matching for smt solvers. In F. Pfenning, editor, *CADE*, volume 4603 of *Lecture Notes in Computer Science*, pages 183–198. Springer, 2007.

[23] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 3 — A computer algebra system for polynomial computations, 2010. http://www.singular.uni-kl.de.

[24] M.-L. Dequéant, S. Ahnert, H. Edelsbrunner, T. M. A. Fink, E. F. Glynn, G. Hattem, A. Kudlicki, Y. Mileyko, J. Morton, A. R. Mushegian, L. Pachter, M. Rowicka, A. Shiu, B. Sturmfels, and O. Pourquié. Comparison of pattern detection methods in microarray time series of the segmentation clock. *PLoS ONE*, 3(8):e2856, 08 2008.

[25] M.-L. Dequéant, E. Glynn, K. Gaudenz, M. Wahl, J. Chen, A. Mushegian, and O. Pourquié. A Complex Oscillating Network of Signaling Genes Underlies the Mouse Segmentation Clock. *Science*, 314(5805):1595–1598, 2006.

[26] M. L. Dequeant, J. Morton, L. Pachter, O. Pourquie, A. Shiu, B. Sturmfels, and O. Wienand. Application of convex rank tests to microarray data. *preprint*, 2006.

[27] P. Diaconis and B. Sturmfels. Algebraic algorithms for sampling from conditional distributions. *Annals of Statistics*, 26:363–397, 1998.

[28] B. Dutertre and L. de Moura. A fast linear-arithmetic solver for dpll(t). *Proc. International Conference Computer Aided Verification (CAV)*, 4144:81–94, 2006.

[29] C. R. Fletcher. Unique factorization rings. *Proceedings of the Cambridge Philosophical Society*, 65(3):579–583, May 1969.

[30] E. Gawrilow and M. Joswig. polymake: a framework for analyzing convex polytopes. In G. Kalai and G. Ziegler, editors, *Polytopes — Combinatorics and Computation*, pages 43–74. Birkhäuser, 2000.

[31] E. Gawrilow and M. Joswig. polymake: an approach to modular software design in computational geometry. In *Proceedings of the 17th Annual Symposium on Computational Geometry*, pages 222–231. ACM, 2001. June 3-5, 2001, Medford, MA.

[32] D. Geiger, C. Meek, and B. Sturmfels. On the toric algebra of graphical models. *Annals of Statistics*, 34:1463–1492, 2006.

[33] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. On an implementation of standard bases and syzygies in singular. *Reports on Computer Algebra (ZCA Report)*, 2000.

[34] D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at `http://www.math.uiuc.edu/Macaulay2/`, 2002.

[35] G.-M. Greuel and G. Pfister. *A Singular Introduction to Commutative Algebra*. Springer, Berlin and Heidelberg, 2002.

[36] G.-M. Greuel, G. Pfister, and H. Schönemann. *Symbolic computation and automated reasoning, The Calculemus-2000 Symposium*, chapter SINGULAR 3.1 — A computer algebra system for polynomial computations, pages 227–233. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[37] G.-M. Greuel, F. Seelisch, and O. Wienand. The groebner basis of the ideal of vanishing polynomials. *Journal of Symbolic Computation (In Press, Corrected Proof)*, ISSN: 0747-7171:DOI: 10.1016/j.jsc.2010.10.006, 2010.

[38] L. Halbeisen and N. Hungerbühler. Powers and polynomials in F_m. *Elemente der Mathematik*, 54:118–129, 1999. 10.1007/s000170050003.

[39] R. Hemmecke, R. Hemmecke, and P. Malkin. 4ti2 version 1.2—computation of Hilbert bases, Graver bases, toric Gröbner bases, and more., September 2005.

[40] R. Hemmecke, J. Morton, A. Shiu, B. Sturmfels, and O. Wienand. Three counterexamples on semigraphoids. *Combinatorics, Probability and Computing*, 17:239–257, 2008.

[41] R. Hemmecke, A. Takemura, and R. Yoshida. Computing holes in semi-groups. *Mathematical Engineering Technical Reports*, 46, 2006.

[42] H. Hirai. Sequences of stellar subdivisions. *Preprint*, 2006.

[43] N. Hungerbühler and E. Specker. A generalization of the smarandache function. *Integers: Electronic J. Combinatorial Number Theory*, 6:A23, 2006.

[44] A. J. Kempner. Polynomials and their residue systems. *Amer. Math. Soc. Trans.*, 22:240–266, 1921.

[45] D. Kroening and S. A. Seshia. Formal verification at higher levels of abstraction. *ICCAD '07 Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 572–578, 2007.

[46] L. Lovász. *Mathematical Programming: The State of the Art*, chapter Submodular functions and convexity, pages 235–257. Springer, 1983.

[47] F. Matus. Ascending and descending conditional independence realtions. *Information Theory, Statistical Decision Functions and Random Processes, Transactions of 11th Prague Conference (S Kubik, J A Visek eds)*, B, 1992.

[48] F. Matúš. On equivalence of markov properties over undirected graphs. *Journal of Applied Probability*, 29:745–749, 1992.

[49] F. Matúš. Conditional independences among four random variables. iii. final conclusion. *Combin. Probab. Comput.*, 8:269–276, 1999.

[50] F. Matúš. Conditional probabilities and permutohedron. *Ann. Inst. H. Poincaré Probab. Statist.*, 39:687–701, 2003.

[51] F. Matúš. Towards classification of semigraphoids. *Discrete Mathematics*, 277:115–145, 2004.

[52] F. Matúš. Towards classification of semigraphoids. *Discrete Mathematics*, 277:115–145, 2004.

[53] E. Miller and B. Sturmfels. Combinatorial commutative algebra. In *Graduate Texts in Mathematics*. Springer Verlag, New York, 2004.

[54] J. Morton, L. Pachter, A. Shiu, B. Sturmfels, and O. Wienand. Geometry of rank tests. probabilistic graphical models (pgm 3). *Proceedings of the conference Probabilistic Graphical Models (PGM 3), Prague, Czech Republic*, page 8pp., 2006.

[55] J. Morton, L. Pachter, A. Shiu, B. Sturmfels, and O. Wienand. Convex rank tests and semigraphoids. *SIAM Journal on Discrete Mathematics*, 23 (3):1117–1134, 2009.

[56] J. Morton, A. Shiu, L. Pachter, and B. Sturmfels. The cyclohedron test for finding periodic genes in time course expression studies. *Statistical Applications in Genetics and Molecular Biology*, 6:1–21, 2007.

[57] G. H. Norton and A. Salagean. Strong gröbner bases for polynomials over a principal ideal ring. *Bulletin of the Australian Mathematical Society*, 64:505–528, August 2001.

[58] G. H. Norton and A. Salagean. Gröbner bases and products of coefficient rings. *Bulletin of the Australian Mathematical Society*, 65(1):145–152, 2002.

[59] Onespin Solutions GmbH Munich, Germany. `www.onespin.com`.

[60] F. Pauer. Gröbner bases with coefficients in rings. *Journal of Symbolic Computation*, 42(11-12):1003–1011, 2007.

[61] E. Pavlenko, M. Wedler, D. Stoffel, W. Kunz, A. Dreyer, F. Seelisch, and G.-M. Greuel. STABLE: a new qf-bv smt solver for hard verification problems combining boolean reasoning with computer algebra. In *Automation and Test in Europe (DATE)*, March 2011, Grenoble, France. accepted, to appear.

[62] E. Pavlenko, M. Wedler, D. Stoffel, O. Wienand, E. Karibaev, and W. Kunz. Modeling of custom-designed arithmetic components in abl normalization. *Proc. Forum on Specification & Design Languages (FDL)*, 2008.

[63] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publisher, San Mateo CA, 1988.

[64] E. J. G. Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

[65] A. Postnikov. Permutohedra, associahedra, and beyond. *arXiv*, 2005.

[66] A. Postnikov, V. Reiner, and L. Williams. Faces of simple generalized permutohedra. *Preprint*, 2006.

[67] G. Pruesse and F. Ruskey. Generating linear extensions fast. *SIAM Journal on Computing*, 23(2):373–386, 1994.

[68] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[69] S. A. Seshia, S. K. Lahiri, and R. E. Bryant. A hybrid SAT-based decision procedure for separation logic with uninterpreted functions. *DAC '03 Proceedings of the 40th annual Design Automation Conference*, pages 425 – 430, 2003.

[70] N. Shekhar, P. Kalla, and F. Enescu. Equivalence verification of arithmetic datapath with multiple word-length operands. *DATE '06 Proceedings Design, Automation and Test in Europe*, page 6 pp., 2006.

[71] N. Shekhar, P. Kalla, and F. Enescu. Equivalence verification of polynomial datapaths using ideal membership testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26, Issue: 7:1320–1330, 2007.

[72] N. Shekhar, P. Kalla, F. Enescu, and S. Gopalakrishnan. Equivalence verification of polynomial datapaths with fixed-size bit-vectors using finite ring algebra. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 291–296, Washington, DC, USA, 2005. IEEE Computer Society.

[73] N. Shekhar, P. Kalla, F. Enescu, and S. Gopalakrishnan. Equivalence verification of polynomial datapaths with fixed-size bit-vectors using finite ring algebra. *Proc. International Conference on Computer-Aided Design (ICCAD) International Conference on Computer-Aided Design ICCAD-2005. IEEE/ACM*, pages 291 – 296, 2005.

[74] D. Singmaster. On polynomial functions (mod m). *Journal of Number Theory*, 6(5):345–352, October 1974.

[75] F. Smarandache. A function in the number theory. *Analele Univ. Timisoara, Fascicle 1*, XVIII:79–88, 1980.

[76] D. J. Smith. Verilog compared & contrasted – plus modeled example written in VHDL. *Verilog and C, in: DAC '96: Proceedings of the 33rd annual conference on Design automation, ACM Press*, pages 771–776, 1996, NY, USA.

[77] R. P. Stanley. *Enumerative combinatorics. Vol. 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1997. With a foreword by Gian-Carlo Rota, Corrected reprint of the 1986 original.

[78] J. Stembridge. Maple packages for symmetric functions, posets, root systems, and finite coxeter groups. Available at `www.math.lsa.umich.edu/~jrs/maple.html`.

[79] M. Studený. Conditional independence relations have no finite completeness characterization. *Kybernetika*, 25:72–79, 1990.

[80] M. Studený. *Probablistic conditional independence structures*. Springer Series in Information Science and Statistics. Springer-Verlag, London, 2005.

[81] M. Studený, R. R. Bouckaert, and T. Kocka. Extreme supermodular set functions over five variables. *Institute of Information Theory and Automation, Research report n. 1977*, 2000, Prague.

[82] The OEIS Foundation Inc. The on-line encyclopedia of integer sequences, sequence a046873. published electronically at http://oeis.org/A046873, 2010.

[83] The OEIS Foundation Inc. The on-line encyclopedia of integer sequences, sequence a046873. published electronically at http://oeis.org/A046873, 2010.

[84] J. Tits. Le problème des mots dans les groupes de coxeter. *Symposia Math.*, 1:175–185, 1968.

[85] G. Van Rossum. *The Python Language Reference Manual*. Network Theory Ltd., September 2003.

[86] Y. Watanabe, N. Homma, T. Aoki, and T. Higuchi. Application of symbolic computer algebra to arithmetic circuit verification. *ICCD 2007 - 25th International Conference on Computer Design*, pages 25 – 32, 2007.

[87] M. Wedler, S. Stoffel, R. Brinkmann, and W. Kunz. A normalization method for arithmetic data-path verification. *IEEE Transactions on Computer-Aided Design*, 26 (11):1909–1922, 2007.

[88] D. Weinreich. The rank ordering of genotypic fitness values predicts genetic constraint on natural selection on landscapes lacking sign epistasis. *Genetics*, 171:1397–1405, 2005.

[89] O. Wienand, M. Wedler, D. Stoffel, W. Kunz, and G.-M. Greuel. An algebraic approach for proving data correctness in arithmetic data paths. *CAV '08: Proceedings of the 20th international conference on Computer Aided Verification*, pages 473–486, July 2008, Princeton, NJ, USA.

[90] K. Willbrand, F. Radvanyi, J. P. Nadal, J. P. Thiery, and T. Fink. Identifying genes from up-down properties of microarray expression series. *Bioinformatics*, 21(20):3859–3864, 2005.

[91] O. Zariski and P. Samuel. *Commutative Algebra I*, volume 28 of *Graduate Texts in Mathematics*. Springer, 1979.

[92] G. M. Ziegler. *Lectures on polytopes, Graduate Texts in Mathematics*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1998.

List of Figures

List of Tables

Index

Wissenschaftlicher Werdegang

2000	Abitur (Herzog-Johann-Gymnasium, Simmern, Rheinland-Pfalz)
2000 - 2001	Früheinstieg in das Physikstudium an der Universität Kaiserslautern
2001 - 2004	Studium der Mathematik mit Nebenfach Informatik an der Universität Kaiserslautern
2002	Diplomvorprüfung in Mathematik (Universität Kaiserslautern)
2004	Diplom in Mathematik (Technische Universität Kaiserslautern)
seit 2004	Doktorand bei Prof. Dr. G.-M. Greuel an der Technischen Universität Kaiserslautern
2006 - 2008	Wissenschaftlicher Mitarbeiter am Fachbereich Mathematik der Technischen Universität Kaiserslautern