Computations of Binomial Ideals

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

by

Deepanjan Kesh

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

February, 2012

CERTIFICATE

It is certified that the work contained in the thesis entitled "*Computations of Binomial Ideals*", by "*Deepanjan Kesh*", has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Shashank K Mehta)
Professor,
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

February, 2012

Synopsis

Consider the polynomial ring $k[x_1, \ldots, x_n]$, where k is a field. A **binomial** in such a ring is a polynomial of the form

$$c \cdot \mathbf{x}^{\alpha} + d \cdot \mathbf{x}^{\beta},$$

where $c, d \in k$ and $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$. A binomial of the form

$$\mathbf{x}^{\alpha} - \mathbf{x}^{\beta}$$

is called a **pure difference** binomial. An ideal in the polynomial ring $k[x_1, \ldots, x_n]$ which has a generating set comprising only of binomials is called a **binomial ideal**. If a basis has only pure difference binomials, then the ideal is called a **pure difference binomial ideal**. In this thesis, we will be concerned with the computations of various binomial ideals.

One of the most useful ideas in computational commutative algebra is the notion of Gröbner basis of an ideal in a polynomial ring $k[x_1, \ldots, x_n]$. Most algorithms in commutative algebra are based on the computation of Gröbner bases of ideals, for example equality of ideals, ideal membership, intersection of ideals, elimination ideals, computing varieties (CLO07; AL94). In most cases, the computational cost of the problem is dominated by the computational cost of these bases. Computation of Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring (MM82). This suggests that computations, if possible, should be delegated to rings of fewer variables.

This idea has been exploited in the computation of toric ideal by Hemmecke and Malkin (HM09). Computation of toric ideals, which are a sub-class of pure difference binomial ideals, involve the computation of saturation. There are several well known algorithms to compute toric ideals (HS95; CT91; BSR99). In all of these algorithms, all Gröbner basis computations are performed in the original ring $k[x_1, \ldots, x_n]$, to which the ideal belongs. Hemmecke and Malkin proposed the *Project and Lift* algorithm in which bulk of the computation is performed in rings of lesser number of variables, namely, $k[x_1, \ldots, x_i]$. In their approach , they use the projection map $\pi : k[x_1, \ldots, x_n] \to k[x_1, \ldots, x_i]$ given by $\pi(f) = f|_{x_{i+1}=1,\ldots,x_n=1}$. In order to lift the ideals back to the original ring it is essential that π induces an isomorphism of the relevant class of ideals in the two rings. Their algorithm locates situations, if any, where such isomorphism exists. There it maps the ideal to an ideal in the lower ring, computes its saturation and lifts it back to the original ring.

In this thesis, motivated by *Project and Lift* algorithm, we develop new projection homomorphisms and apply it to a variety of computations.

In Chapter 2 of the thesis, we present an algorithm for computing toric ideals where, unlike *Project and Lift*, we symbolically project the ideal to $k[x_1, \ldots, x_i]$. This in turn amounts to the computation of one Gröbner basis in $k[x_1, \ldots, x_i]$ for each i. This symbolic projection allows us to compute the saturation of all pure-difference binomial ideals, not just toric ideals.

In Chapter 3, we further develop the idea of projection into rings with lesser number of variables using a more sound approach based on *localization*. The localization of polynomial rings in our case leads to rings which are polynomial rings over localized rings. As Gröbner basis is not defined for ideals in such rings, we propose the concept of *pseudo Gröbner basis* for binomial ideals in these rings. We also adopt Buchberger's algorithm to compute *pseudo Gröbner basis* and generalize a crucial result about Gröbner basis to pseudo Gröbner basis. Using this machinery, we devise a saturation algorithm for homogeneous binomial ideals (not just pure difference binomial ideals).

A Divide and Conquer Method

In Chapter 4, we further extend the idea of projecting into rings of fewer variables and propose a general framework to a variety of computation related to binomial ideals. We propose a *divide-and-conquer* technique to solve the computational problems in the domain of binomial ideals.

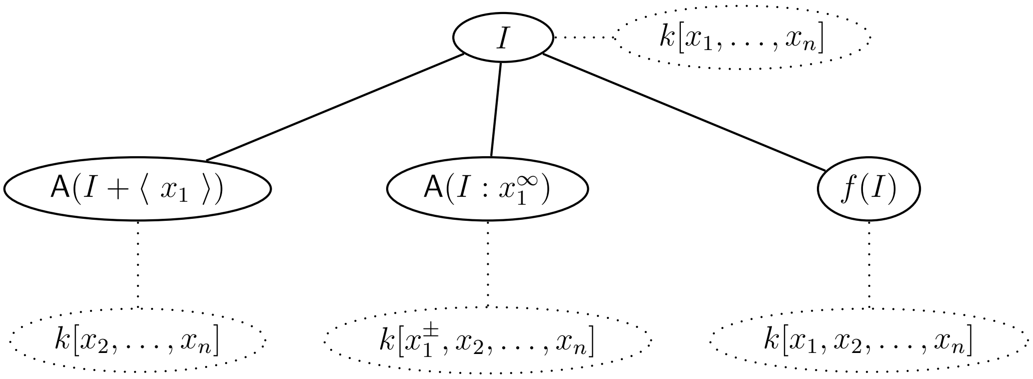
I
$k[x_1, \dots, x_n]$
$\mathsf{A}(I + \langle\ x_1\ \rangle)$
$\mathsf{A}(I : x_1^\infty)$
$f(I)$
$k[x_2, \dots, x_n]$
$k[x_1^\pm, x_2, \dots, x_n]$
$k[x_1, x_2, \dots, x_n]$

Figure 1: Reducing the problem in smaller rings.

The essence of the strategy has been described in Figure 1.1. Consider the polynomial ring $R[x_1, \dots, x_n]$, a binomial ideal $I \subseteq R[x_1, \dots, x_n]$ and $\mathsf{A}(I)$ denotes the object to be computed. Here R is a Laurent polynomial ring. The pseudo Gröbner bases are well defined in $R[x_1, \dots, x_n]$. As the figure suggests, we reduce the problem into three subproblems –

$\mathsf{A}\,(\mathbf{I} + \langle\ \mathbf{x_1}\ \rangle)$ – Ideal $I + \langle\ x_1\ \rangle$ is mapped into the ring $R[x_2, \dots, x_n]$ by the natural modulo map from $R[x_1, \dots, x_n] \to R[x_2, \dots, x_n]$, the computations are performed in this smaller ring, and the solution is mapped back to the parent ring.

$\mathsf{A}\,(\mathbf{I} : \mathbf{x_1^\infty})$ – Ideal $I : x_1^\infty$ is mapped into the ring $R[x_1^\pm, x_2, \dots, x_n]$. The ± sign over the variable x_1 denotes that we allow negative indices for x_1. For the purposes of the computations involved, we will be treating the ring as polynomial ring in variables $\{x_2, \dots, x_n\}$ over the Laurent polynomial ring $R' = R[x_1^\pm]$.

$\mathbf{f}(\mathbf{I})$ – This subproblem is to be solved in the original ring $R[x_1, \dots, x_n]$, where the function f depends on the problem we are tackling. This approach becomes effective only if $f(I)$ computation does not involve the computation of a Gröbner basis.

Solutions of these subproblems are lifted to the original ring and combined to compute the solution of the original problem. This combination step depends on the problem under consideration. The first two subproblems are solved recursively.

In this thesis, we have applied this framework on the following four problems – radical, minimal primes, cellular decomposition, and saturation of binomial ideals.

Acknowledgement

I am deeply indebted to my thesis supervisor Prof. Shashank K Mehta for his guidance throughout my Ph.D. tenure. His enthusiasm and sincerity is infectious. Almost all of the work that have been carried out towards this thesis has been the result of my discussions with him. He has spent countless hours with me whenever I was bereft of ideas and I was not making any discernible progress towards my thesis. His guidance and generosity was not restricted to the academic sphere, but he has also lend an immense helping hand in my social life. I will forever be indebted to him.

I would also like to express my gratitude towards my family - my mother for her ceaseless, sometimes if not foolish, optimism; my father for his relentless support; and my sister forever believing in me. It would be a fool's errand to describe the role they have played in my life, because no matter how much I try I cannot do justice. My only wish is to justify their support and try to live up to their expectations.

I, humbly, would like to thank Prof. Sumit Ganguly for introducing me to the world of scientific research and for allowing me to work with him resulting in my first academic publication. I wish I could have worked harder to repay his faith in me. I would also like to thank various faculties of our department including Prof. Somenath Biswas and Prof. Manindra Agrawal for instilling the right kind of attitude towards problem solving and research in general.

Life as a Ph.D. scholar would be an ordeal if not for the presence of one's friends. I was fortunate enough to be blessed with a myriad of friends, and I would like to thank all of them for their support. I would especially like to thank Ramprasad Saptharishi, Suman Guha and Chandan Saha for always being there for me, and for the countless hours of counselling I received from them. I would also take this

oppurtunity to thank Sagarmoy Dutta for his honesty which has always helped me find perspective, when I needed it the most.

I would also like to thank Microsoft Research India and IMPECS for funding parts of my Ph.D. programme. I would also like to thank Department of Computer Science and Engineering at IIT Kanpur for funding my trips to attend conferences which has helped me immensely in my thesis work.

To my family

Contents

Chapter 1

Introduction

Consider the polynomial ring $k[x_1, \ldots, x_n]$, where k is a field. A **binomial** in such a ring is a polynomial of the form

$$c \cdot \mathbf{x}^{\alpha} + d \cdot \mathbf{x}^{\beta},$$

where $c, d \in k$ and $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$. An ideal in the polynomial ring $k[x_1, \ldots, x_n]$, which has a generating set comprising only of binomials, is called a **binomial ideal** . In this thesis, we will be concerned with the computations of various binomial ideals.

1.1 Why Study Binomial Ideals?

Binomial ideals, unlike general polynomial ideals, possess rich combinatorial structure which can be exploited while computing various structures derived from them, for example Gröbner bases, primary decomposition, and associated primes (Tho95; ES96; Kah10). **Pure difference binomials** are binomials of the form $\mathbf{x}^{\alpha} - \mathbf{x}^{\beta}$. The varieties of pure difference prime binomial ideals are exactly the toric varieties. Hence, such ideals are also known as toric ideals (Ful93). Moreover, quotients of polynomial rings by pure difference binomial ideals form commutative semigroup rings (Gil84). There is a large literature studying applications and computations of toric ideals (Stu95; BSR99).

Apart from a purely academic interest in the subject of binomial ideals, their study is also motivated by the fact that they are often encountered in interesting problems in diverse fields. These include solving integer programs (HS95; CT91;

UWZ97a; TW97), computing primitive partition identities (Stu95, Chapters 6,7), and solving scheduling problems (TTN95). In algebraic statistics, closures of discrete exponential families have been identified with nonnegative toric varieties (GMS06). Primary decomposition of binomial ideals enter algebraic statistics while modelling conditional independences among random variables (DSS09).

The theory of binomial ideals was developed in a seminal paper by Eisenbud and Sturmfels (ES96). Their paper not only showed various properties of binomial ideals – for example, the radicals and associated primes of binomial ideals are themselves binomial ideals – but they also show how to compute these structures.

1.2 Focus of the thesis

One of the most useful ideas in computational commutative algebra is the notion of Gröbner basis of an ideal in a polynomial ring, say $k[x_1, \dots, x_n]$. It has found many applications in computations related to these ideals – equality of ideals, ideal membership, intersection of ideals, elimination ideals, computing varieties, to name a few (CLO07; AL94). Presently every non-trivial algorithm for computation of ideals is based on the computation of some Gröbner basis. The first and perhaps the most popular algorithm to compute a Gröbner basis is due to Buchberger (Buc76). Recently, Faugére (Fau99; Fau02) has presented much faster algorithms to compute Gröbner bases. A more detailed discussion of the properties of Gröbner bases and the Buchberger algorithm can be found in Appendix B.

We now state two crucial observations which motivated this thesis –

- Most of the computations involving binomial ideals compute one or more Gröbner bases (ES96), and

- Any algorithm to compute Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring. (MM82)

So, it would seem judicious if part of the computations can be done in rings having smaller number of variables, and use this result to arrive at a solution for the original problem.

This idea has been exploited in the computation of toric ideal by Hemmecke and Malkin (HM09). Computation of toric ideals, which are a subclass of pure difference binomial ideals, involve the computation of saturation. There are several well known algorithms to compute toric ideals (HS95; CT91; BSR99). In all of these algorithms, all Gröbner basis computations are performed in the original ring $k[x_1, \ldots, x_n]$, to which the ideal belongs. Hemmecke and Malkin (HM09) proposed the *Project and Lift* algorithm in which bulk of the computation is performed in rings of lesser number of variables, namely, $k[x_1, \ldots, x_i]$. In their approach, they use the projection map $\pi : k[x_1, \ldots, x_n] \rightarrow k[x_1, \ldots, x_i]$ given by $\pi(f) = f|_{x_{i+1}=1,\ldots,x_n=1}$. In order to lift the ideals back to the original ring it is essential that π induces an isomorphism of a relevant class of ideals. Their algorithm locates situations, if any, where such isomorphism exists. There it reduces the ideal to a lower ring, computes its saturation and lifts it back to the original ring.

In this thesis, motivated by *Project and Lift* algorithm, we develop new projection homomorphisms which can be applied to the computation of a variety of binomial ideals.

In Chapter 2 of the thesis, we present an algorithm for computing toric ideals where, unlike *Project and Lift*, we symbolically project the ideal to $k[x_1, \ldots, x_i]$. This in turn amounts to the computation of one Gröbner basis in $k[x_1, \ldots, x_i]$ for each i. While the algorithm due to Hemmecke and Malkin (HM09) is specifically designed to compute toric ideals, our algorithm can compute saturation of arbitrary pure difference binomial ideals.

In Chapter 3, we further develop the idea of projection onto rings with lesser number of variables using a more sound approach based on *localization* of polynomial ring. As Gröbner basis is not defined for ideals in such rings, we propose the concept of *pseudo Gröbner basis* for binomial ideals in localized polynomial rings. An algorithm to compute the saturation of homogeneous binomial ideals is proposed based on pseudo Gröbner basis.

In the final chapter, a general framework is proposed for a *divide and conquer* based algorithm in which a problem on i-variable polynomial ring is reduced to problems in $(i-1)$-variable polynomial rings. We apply this approach to compute radical, prime decomposition, and cellular decomposition of a binomial ideal.

1.3 Computing Toric Ideals

When it comes to applications, toric ideals are by far the most useful of all binomial ideals. They are used in model selection tasks and integer programming (Tho95). The applications of binomial ideals that we have seen earlier, like computing primitive partition identities, and solving scheduling problem have all to do with toric ideals.

1.3.1 Problem Statement

Let $A \in \mathbb{Z}^{m\times n}$ be an integer matrix –

$$A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{bmatrix}.$$

The lattice kernel of such a matrix is defined as –

$$\ker A \triangleq \{ \ \mathbf{u} \in \mathbb{Z}^n \mid A \cdot \mathbf{u} = 0 \ \}.$$

i.e., integer solutions of $A\mathbf{u} = 0$. For any $\mathbf{u} \in \ker A$, we further define the vectors $\mathbf{u}_+$ and $\mathbf{u}_-$ as –

$$\mathbf{u}_+[i] \triangleq \begin{cases} \mathbf{u}[i], & \mathbf{u}[i] > 0 \\ 0, & \text{otherwise} \end{cases}$$
$$\mathbf{u}_- \triangleq \mathbf{u}_+ - \mathbf{u}$$

The **toric ideal** of a matrix A, denoted by I_A is defined to be the ideal

$$I_A \triangleq \langle \ \{ \ \mathbf{x}^{u_+} - \mathbf{x}^{u_-} \mid \mathbf{u} \in \ker A \ \} \ \rangle.$$

Here, $\mathbf{x}^v$ for any non-negative integer vector $\mathbf{v} \in \mathbb{Z}^n_{\geq 0}$, is the monomial $x_1^{v[1]} x_2^{v[2]} \cdots x_n^{v[n]}$.
Pure difference binomials are binomials of the form

$$\mathbf{x}^{\alpha} - \mathbf{x}^{\beta}.$$

So, toric ideals are pure difference binomial ideals. It was shown in (ES96), Corollary 2.2 that over algebraically closed field, toric ideals are also prime.

Generating sets of toric ideals are known as “Markov Bases” in statistics. Chapter 2 addresses the problem of computing a generating set of I_A, which we loosely call the problem of computing a toric ideal.

1.3.2 Solution

Suppose V is a lattice kernel basis, i.e., a basis of $\ker A$ which generates the kernel vectors with integer coefficients. Let J_V be the ideal

$$J_V = \langle \; \{ \; \mathbf{x}^{u_+} - x^{u_-} \mid \mathbf{u} \in V \; \} \; \rangle.$$

It is easy to show that (Stu95, Chapter 4)

$$I_A = \left\{ \; f \in k[x_1, \ldots, x_n] \mid \mathbf{x}^\alpha f \in J_V, \alpha \in \mathbb{Z}^n_{\geq 0} \; \right\}.$$

The set on the right hand side is an ideal which is called the **saturation** of J_V with respect to all the variables in the ring $k[x_1, \ldots, x_n]$, and is defined as

$$J_V : (x_1 \cdots x_n)^\infty \triangleq \left\{ \; f \in k[x_1, \ldots, x_n] \mid \mathbf{x}^\alpha f \in J_V, \alpha \in \mathbb{Z}^n_{\geq 0} \; \right\}.$$

Then $I_A = J_V : (x_1 \cdots x_n)^\infty$.

We see that the computation of a toric ideal has two steps: computation of lattice kernel basis, V and the saturation of J_V. The first step has a polynomial time solution by computing the **Hermite normal form** of A (KB79; CC82). The more complicated and expensive step is the saturation computation.

1.3.2.1 Previous work

An early algorithm to compute I_A involved computation of a Gröbner basis in a polynomial ring of $m + n + 1$ variables (Stu95, Chapter 4), where A is the $m \times n$ matrix.

An algorithm for saturation, working in n variables, is due to Biase and Urbanke (BU95). It transforms the matrix A to another matrix A' by negating some columns such that one of the rows has all non-negative entries. If V' is the lattice basis of A', then they have shown that $I_{A'} = J_{V'}$, i.e. no saturation is required. Now, to compute the original ideal, they replace one negated column at a time by the original one and compute the toric ideal for the corresponding matrix from the generating set of the previous matrix. Each step involves the computation of one Gröbner basis. Another algorithm which also works in n variables is due to Sturmfels (HS95; Stu95). It computes the toric ideal iteratively, computing the saturation with x_i in the i-th iteration. Each iteration involves the computation of one Gröbner basis. The performances of the two algorithms are comparable, see (HS95). Bigatti et. al. (BSR99) parallelized the Sturmfels' algorithm.

As mentioned earlier, Hemmecke and Malkin (HM09) presented an entirely new approach called *Project and lift*. Given $\sigma \subseteq \{1, \ldots, n\}$, they define a projective map

$$\pi_\sigma : \mathbb{Z}^n \to \mathbb{Z}^{|\overline{\sigma}|}$$

by setting components in σ to 1. Here, $\overline{\sigma}$ is the set $\{1, \ldots, n\} \setminus \sigma$. Let $\mathcal{L}$ be the lattice generated by $\ker A$. Their algorithm starts with computing a set σ such that

$$\ker \pi_\sigma \bigcap \mathcal{L} = \{0\} \text{ and } \mathcal{L}^\sigma \bigcap \mathbb{N}^{|\overline{\sigma}|} = \{0\}\,.$$

The algorithm then perform $|\overline{\sigma}|$ Gröbner basis computations in a ring with $|\overline{\sigma}|$ variables and one Gröbner basis computation each in rings with variables $|\overline{\sigma}| + 1, |\overline{\sigma}| + 2, \ldots, n$, respectively. As it is evident, the bulk of the computation is performed in rings having less than n variables.

1.3.2.2 Our Approach

We present an algorithm that requires the computation of one Gröbner basis in $k[x_1, \ldots, x_i]$ for each i. Unlike *Project and Lift*, we symbolically project the ideal to $k[x_1, \ldots, x_i]$.

While the algorithms due to Biase-Urbanke (BU95) and Hemmecke-Malkin (HM09) is specifically designed to compute toric ideals, our algorithm can compute saturation of arbitrary pure difference binomial ideals. On the other end of the spectrum, Sturmfels' algorithm is less efficient but it can compute saturation of arbitrary polynomial ideal.

1.4 Saturating Binomial Ideal

This problem finds applications in computing the radicals, minimal primes, cellular decompositions, etc., of a homogeneous binomial ideal, see (ES96). As observed earlier, it is also the key step in the computation of a toric ideal. Chapter 3 is devoted to this problem.

1.4.1 Problem Description

Let

$$b = c\mathbf{x}^{\alpha} + d\mathbf{x}^{\beta}$$

be a binomial, and $\vec{d} \in \mathbb{Z}^n_{\geq 0}$ be a vector. b is a said to be **homogeneous** w.r.t. $\vec{d}$, if

$$\vec{d} \cdot \alpha = \vec{d} \cdot \beta.$$

Vector $\vec{d}$ is called the **grading vector**. An ideal with at least one homogeneous binomial basis is called a homogeneous binomial ideal.

We describe a fast algorithm to compute the saturation, $I : (x_1 \cdots x_n)^{\infty}$, of a homogeneous binomial ideal I. Every binomial ideal in a n-variable polynomial ring can be "homogenized" using an additional variable.

1.4.2 Solution

There are algorithms to compute the saturation of any ideal in $k[x_1, \ldots, x_n]$ (not just binomial ideals). One such algorithm is described in exercise 4.4.7 in (CLO07). It involves a Gröbner basis computation in $n+1$ variables. Another solution is due to Sturmfels (Stu95) which involves n Gröbner basis computations in n variables.

Our approach is the same as in the previous case, doing bulk of our computation in rings with less number of variables compared to the original ring. In this case, we propose a more sound approach to project an ideal to a ring of lesser number of variables using localization. We also propose the concept of *pseudo Gröbner basis* for binomial ideals in localized rings. This generalization of Gröbner bases is essential for our saturation algorithm.

1.5 A General Framework

In Chapter 4, we extend the ideas of the previous two chapters and propose a general framework to compute several binomial ideals. We restate the two crucial observations behind this work

- most of these computations involve computing Gröbner basis of some ideals, and
- Buchberger's algorithm to compute Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring.

In light of these observations, we propose a *divide-and-conquer* technique to solve the computational problems in the domain of binomial ideals. We apply this technique to the computation of saturation, radical, minimal primes, and cellular decomposition of binomial ideals.

The essence of the strategy has been described in Figure 1.1. Consider the polynomial ring $R[x_1, \ldots, x_n]$, a binomial ideal $I \subseteq R[x_1, \ldots, x_n]$ and $\mathsf{A}(I)$ denotes the object to be computed. As the figure suggests, we divide the problem into three subproblems –

$\mathsf{A}\left(\mathbf{I} + \langle\ \mathbf{x_1}\ \rangle\right)$ – This ideal is mapped onto the ring $R[x_2, \ldots, x_n]$ by the natural modulo map from $R[x_1, \ldots, x_n] \to R[x_2, \ldots, x_n]$, the computations are performed in this smaller ring, and the solution is mapped back onto the parent ring.

$\mathsf{A}\left(\mathbf{I} : \mathbf{x_1^\infty}\right)$ – This ideal is mapped onto the ring $R[x_1^{\pm}, x_2, \ldots, x_n]$. The ± sign over the variable x_1 denotes that we allow negative indices for x_1. For the purposes of the computations involved, we will be treating the ring as polynomial ring in variables $\{x_2, \ldots, x_n\}$ over the Laurent ring $R[x_1^{\pm}]$. As we will see, the most expensive computation in this ring is *pseudo Gröbner basis* and it involves one less variable.

$\mathbf{f}(\mathbf{I})$ – This subproblem is to be solved in the original ring $R[x_1, \ldots, x_n]$, where the function f depends on the problem we are tackling. This approach becomes effective only if $f(I)$ computation does not involve the computation of any Gröbner basis.

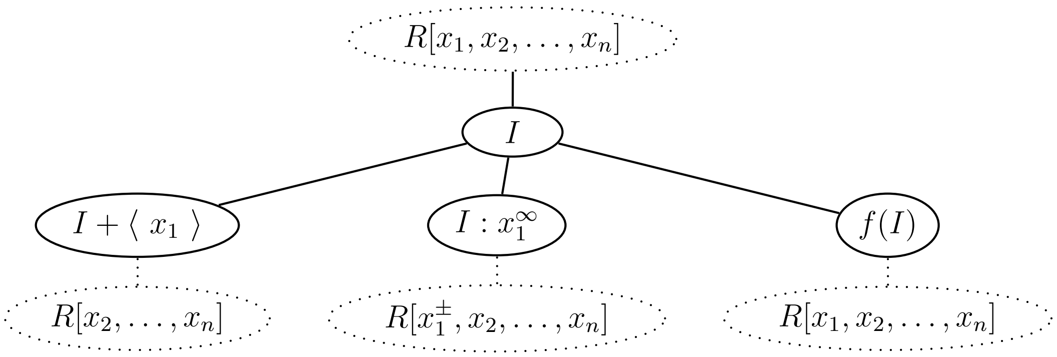
$R[x_1, x_2, \ldots, x_n]$
I
$I + \langle\ x_1\ \rangle$
$I : x_1^\infty$
$f(I)$
$R[x_2, \ldots, x_n]$
$R[x_1^\pm, x_2, \ldots, x_n]$
$R[x_1, x_2, \ldots, x_n]$

Figure 1.1: Reducing the problem in smaller rings.

Solutions of these subproblems are lifted to the original ring and combined to compute the solution of the original problem. This combination step depends on the problem under consideration. The first two subproblems are solved recursively.

In this thesis, we have applied this framework on the following four problems –

Radical Given a binomial ideal I in a polynomial ring $k[x_1, \ldots, x_n]$, radical of I, denoted by $\sqrt{I}$, is defined as

$$\sqrt{I} \triangleq \langle\ \{\ f \mid f^m \in k[x_1, \ldots, x_n]\ \}\ \rangle.$$

Minimal Primes Given a binomial ideal I in a polynomial ring $k[x_1, \ldots, x_n]$, compute the set of minimal primes $\mathcal{P}$ such that

$$\sqrt{I} = \bigcap_{P \in \mathcal{P}} P.$$

Cellular decomposition A binomial ideal $I \subseteq k[x_1, \ldots, x_n]$ is **cellular** if in $k[x_1, \ldots, x_n]/I$ every variable is either a nonzero divisor or a nilpotent. We want to compute a set of cellular ideals $\mathcal{C}$ such that

$$I = \bigcap_{C \in \mathcal{C}} C.$$

Saturation This problem is the same as the problem that has been dealt with in the previous chapters – given a homogeneous binomial ideal $I \subseteq k[x_1, \ldots, x_n]$, we want to compute $I : (x_1 \cdots x_n)$.

Chapter 2

Generalized reduction to compute toric ideals

2.1 Introduction

Toric ideals have many applications including solving integer programs (HS95; CT91; UWZ97b; TW97), computing primitive partition identities (Stu95, Chapters 6,7), and solving scheduling problems (TTN95).

As described earlier, the key step in the computation of a toric ideal involves the saturation of a pure difference binomial ideal. Several algorithms are available in the literature for saturation computation. Since it is an NP hard problem, all approaches can only solve relatively small problems. We propose a new approach which improves upon a well known saturation technique. This chapter is based on the work (KM09; KM10),

2.1.1 Problem Description

We have come across the definition of toric ideals in section 1.3. Recall that, for a given matrix $A \in \mathbb{Z}^{m \times n}$, the computation of the toric ideal of A, denoted by I_A, has two steps:

- Computation of a lattice kernel basis of A, which has a polynomial time solution by computing the **Hermite normal form** of A (KB79; CC82), and

- Saturation of a pure difference binomial ideal. This is the more complicated and expensive step is the saturation computation.

In this chapter we will concentrate on the second step, that is, the saturation of a pure difference binomial ideal.

So, given an ideal $I = \langle\ \mathbf{x}^{\alpha_1} - \mathbf{x}^{\beta_1}, \dots, \mathbf{x}^{\alpha_m} - \mathbf{x}^{\beta_m}\ \rangle$, we want to compute a generating set of

$$\langle\ \mathbf{x}^{\alpha_1} - \mathbf{x}^{\beta_1}, \dots, \mathbf{x}^{\alpha_m} - \mathbf{x}^{\beta_m}\ \rangle : (x_1 \dots x_n)^{\infty}.$$

2.2 Surjective ring homomorphism

Let ϕ denote a surjective ring homomorphism from $k[x_1, \dots, x_n]$ to $k[y_1, \dots, y_m]$.

Definition 2.1. Let $S \subseteq k[x_1, \dots, x_n]$ be a set of polynomials. Then we define $\phi(S)$ as -

$$\phi(S) = \{\ \phi(f) \mid f \in S\ \}.$$

Lemma 2.2. *Let $f_1, \dots, f_s \in k[x_1, \dots, x_n]$. Then*

$$\phi(\langle\ f_1, \dots, f_s\ \rangle = \langle\ \phi(f_1), \dots, \phi(f_s)\ \rangle.$$

Proof. Let $f' \in \phi\left(\langle\ f_1, \dots, f_s\ \rangle\right)$. Then $\exists f \in \langle\ f_1, \dots, f_s\ \rangle$ such that $\phi(f) = f'$. So we have

$$\begin{aligned} f &= \sum_i g_i f_i \\ \implies \quad f' = \phi(f) &= \sum_i \phi(g_i)\phi(f_i) \end{aligned}$$

Here $g_1, \dots, g_s \in k[x_1, \dots, x_n]$. Hence, $f' \in \langle\ \phi(f_1), \dots, \phi(f_s)\ \rangle$.

Conversely, let $f' \in \langle\ \phi(f_1), \dots, \phi(f_s)\ \rangle$. Then $\exists g'_1, \dots, g'_s \in k[y_1, \dots, y_m]$ such that

$$\begin{aligned} f' &= \sum_i g'_i \phi(f_i) \\ &= \sum_i \phi(g_i)\phi(f_i) \\ &= \phi(\sum_i g_i f_i) \end{aligned}$$

The last two equalities follow from the fact that ϕ is surjective. Hence $f' \in \phi(\langle f_1, \ldots, f_s \rangle)$. ■

The **kernel** of a homomorphism ϕ, denoted as $\ker \phi$, is

$$\ker \phi = \{ f \in k[x_1, \ldots, x_n] \mid \phi(f) = 0 \} .$$

The first isomorphism theorem states that –

Theorem 2.3 (First isomorphism theorem)**.** *Let R and S be rings, and let $\phi : R \to S$ be a ring homomorphism. Then:*

- *The kernel of ϕ is an ideal of R.*
- *The image of ϕ is a subring of S, and*
- *The image of ϕ is isomorphic to the quotient ring $R/\ker \phi$.*

In particular, if ϕ is surjective then S is isomorphic to $R/\ker \phi$.

From Theorem 2.3, $k[x_1, \ldots, x_n]/\ker \phi$ is isomorphic to $k[y_1, \ldots, y_m]$. We shall denote this isomorphism by Φ.

Let T be a subset of $k[y_1, \ldots, y_m]$. Then we define ϕ^{-1} as –

$$\phi^{-1}(T) = \{ f \in k[x_1, \ldots, x_n] \mid \phi(f) \in T \} .$$

Observation 1. *Let J be an ideal in $k[y_1, \ldots, y_m]$. Then, $\phi^{-1}(J)$ is an ideal. Also, J and $\phi^{-1}(J)/\ker \phi$ are isomorphic.*

Projections are examples of surjective ring homomorphisms. We will use these maps in all of the algorithms discussed in this chapter.

Definition 2.4. Let the set of variables $\{x_1, \ldots, x_n\}$ be denoted by X, and let $X' \subset X$. Then, the map $\phi : k[X] \to k[X \setminus X']$ is said to be a **projection map** where

$$\phi(f) = f|_{x=1, \forall x \in X'} .$$

It is easy to verify that projections are surjective ring homomorphisms. Next, we will define some symbols to denote specific projective maps. We will use π_i to denote the projection $k[x_1, \ldots, x_n] \to k[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$. In other words, in this map, we set x_i to 1. We will use π_z for the projective map $k[x_i, \cdots, x_{i+j}, z] \to k[x_i, \cdots, x_{i+j}]$. Here, we set z to 1. Finally, to denote the projection from $k[x_1, \ldots, x_n]$ to $k[x_{i+1}, \ldots, x_n]$, we will use the symbol Π_i. In this case, we set the variables $x_1, \ldots, x_i$ to 1.

Observation 2. *π_i, π_z and Π_i are surjective ring homomorphisms.*

2.3 Homogeneous polynomials and saturation

2.3.1 Homogenization

Let $f \in k[x_1, \ldots, x_n]$ be a polynomial, and $\vec{d} \in \mathbb{Z}^n_{\geq 0}$ be a vector. We say f is **homogeneous** w.r.t. $\vec{d}$, if for all monomials $\mathbf{x}^\alpha$ appearing with non-zero coefficient in f, $\vec{d} \cdot \alpha$'s are equal. We call the vector $\vec{d}$, the **grading vector**. If f is not homogeneous, then it can be homogenized using an extra variable z. The new polynomial, though homogeneous, will belong to the ring $k[x_1, \ldots, x_n, z]$.

Let $\vec{d} \in \mathbb{N}^{n+1}$ be a 0/1 vector such that $d_{n+1} = 1$. We define a map $h_{\vec{d}} : k[x_1, \ldots, x_n] \to k[x_1, \ldots, x_n, z]$ such that $h_{\vec{d}}(f)$ will be homogeneous with respect to $\vec{d}$ for every $f \in k[x_1, \ldots, x_n]$. Let $f = \sum_i c_{\alpha_i} \mathbf{x}^{\alpha_i} \in k[x_1, \ldots, x_n]$. Consider the polynomial

$$f' = \sum_i c_{\alpha_i} \mathbf{x}^{\alpha_i} z^{b_i} \in k[x_1, \ldots, x_n, z]$$

where b_i's are so chosen that f' is homogeneous with respect to $\vec{d}$. Let m be the largest integer such that $z^m \mid f'$. Then, we define

$$h_{\vec{d}}(f) = \frac{f'}{z^m}.$$

We shall denote $h_{\vec{d}}(f)$ by $\tilde{f}$ when $\vec{d}$ is known from the context. Observe that $\pi_z(\tilde{f}) = f$. If $B = \{f_i\}_i$ is a set of polynomials of $k[x_1, \ldots, x_n]$, then by homogenization of B we would mean the set $\tilde{B} \subseteq k[x_1, \ldots, x_n, z]$ given by $\{\tilde{f_i}\}_i$. An ideal is said to be **homogeneous** with respect to a grading vector $\vec{d}$, if the ideal has a generating set which is homogeneous with respect to $\vec{d}$.

2.3.2 Ideal Saturation

Let R be a ring, $r \in R$ be a non-zero-divisor and $I \subseteq R$ be an ideal. Then,

$$I : r \triangleq \{ \ s \mid sr^n \in I \ \} .$$

The **saturation** of I w.r.t. r is the ideal

$$I : r^\infty \triangleq \left\{ \ s \mid sr^j \in I, \ \text{for some } j \geq 0 \ \right\} .$$

Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal. Saturation of I with respect to $r = x_1 \cdot x_2 \cdots x_i$, $I : (x_1 \cdot x_2 \cdots x_i)^\infty$ is equal to

$$\left\{ \ f \in k[x_1, \ldots, x_n] \mid \mathbf{x}^\alpha f \in I \text{ for some } \alpha \in \mathbb{Z}^n_{\geq 0} \ \right\} ,$$

which is also an ideal. The following observation is immediate from the definition.

Observation 3. $(\ldots((I : x_1^\infty) : x_2^\infty) \ldots) : x_n^\infty = I : (x_1 \cdots x_n)^\infty$.

In general, the computation of $I : x_i^\infty$ is expensive, see Section 4 in Chapter 4 of (CLO07). It involves computing a Gröbner basis in $n+1$ variables. But in a special case when I is a homogeneous ideal, an efficient algorithm to compute $I : x_i^\infty$ is known from Lemma 12.1 of (Stu95). The Sturmfels' algorithm involves computing a Gröbner basis in n variables.

Before going into the details of the algorithm to compute $J : x_i^\infty$, let us define the following notation. Let f be a polynomial, and a be the largest integer such that x_i^a divides f. Then, we denote the quotient of the division of f by x_i^a as $f \div x_i^\infty$. If B is a set of polynomials, then $B \div x_i^\infty$ denotes the set $\{ \ f \div x_i^\infty \mid f \in B \ \}$.

We will define one more notation. This is related to Gröbner basis. Let $\vec{d} \in \mathbb{Z}^n_{\geq 0}$ be a grading vector for polynomials in the ring $k[x_1, \ldots, x_n]$. Then $\prec_{\vec{d},i}$ denotes the graded reverse lexicographic term ordering with $\vec{d}$ as the grading vector and x_i as the least variable. So if $I \in k[x_1, \ldots, x_n]$ is an ideal, then $\mathcal{G}_{\prec_{\vec{d},i}}(I)$ denotes a Gröbner basis of I with respect to $\prec_{\vec{d},i}$.

Sturmfels' lemma follows.

Lemma 2.5. *(Stu95, lemma 12.1) Let $J \subseteq k[x_1, \ldots, x_n]$ be a homogeneous ideal w.r.t. the grading vector $\vec{d}$. Also let $\mathcal{G}_{\prec_{\vec{d},j}}(J) = \{f_i\}_i$. Then $\left\{f_i \div x_j^\infty\right\}_i$ is a Gröbner basis of $J : x_j^\infty$.*

```
Algorithm 2.1: (Sturmfels' Algorithm) Computation of $\langle B \rangle : x_i^\infty$
Data:
    • A finite generating set, $B$, of an ideal $J \subseteq k[x_1, \dots, x_n]$
    • a variable $x_i$
Result: The Gröbner basis of $\langle B \rangle : x_i^\infty$
$\vec{d} \leftarrow \underbrace{(1, \dots, 1)}_{n+1 \text{ components}}$ ;
$\tilde{B} \leftarrow \left\{ \tilde{f} \mid f \in B \right\}$ /* $\subseteq k[x_1, \dots, x_n, z]$ */
Compute $\mathcal{G}_{\prec_{\vec{d},i}}\left(\langle \tilde{B} \rangle\right)$ ;
$G \leftarrow \left\{ f \div x_i^\infty \mid f \in \mathcal{G}_{\prec_{\vec{d},i}}\left(\langle \tilde{B} \rangle\right) \right\}$ ;
return $\pi_z\,(G)$.
```

Algorithm 2.1 computes $J : x_i^\infty$ for arbitrary ideal J using the following lemma. The following is a useful lemma which shows the relation between the projection homomorphism π_i and the saturation of an ideal.

Lemma 2.6. *Let $I \subseteq k[x_1, \dots, x_n]$ be any ideal. Then $\pi_i(I : x_j^\infty) = \pi_i(J) : x_j^\infty$.*

One can verify this lemma from the definitions of saturation ideals and projections.

2.4 Shadow algorithms under a surjective homomorphism

Let I be an ideal in $k[x_1, \dots, x_n]$ and $\phi : k[x_1, \dots, x_n] \to k[y_1, \dots, y_m]$ be a surjective ring homomorphism. We know from Lemma 2.2 that $\phi(I)$ is an ideal in $k[y_1, \dots, y_m]$. In this section, we show how to compute a basis B of I such that $\phi(B)$ is a Gröbner basis of $\phi(I)$.

Let α and β be two vectors in $\mathbb{Z}^n_{\geq 0}$, and let $\alpha[i]$ and $\beta[i]$ denote their i^{th} components, respectively. Then, by $\alpha \vee \beta$, we denote the vector whose i^{th} component is given by -

$$(\alpha \vee \beta)[i] \triangleq \mathsf{max}\left\{\alpha[i], \beta[i]\right\}.$$

This is also called the LCM of α and β.

In this section, we will assume the existence of an oracle which computes any one element h of $\phi^{-1}(m)$ for any monomial $m \in k[y_1, \ldots, y_m]$. With an abuse of the notations, we shall use

$$h \leftarrow \phi^{-1}(m)$$

as a step in the algorithms given below.

2.4.1 Shadow S-polynomial

Let $\prec$ denote a term order in $k[y_1, \ldots, y_m]$. Consider any two polynomials, $h_1, h_2 \in k[y_1, \ldots, y_m]$. Let

$$c_1\mathbf{y}^{\alpha_1} = \mathsf{in}_\prec(h_1), \text{ and } c_2\mathbf{y}^{\alpha_2} = \mathsf{in}_\prec(h_2)$$

be the leading terms of h_1 and h_2, respectively. Define two vectors β_1 and β_2 as –

$$\beta_1 = (\alpha_1 \vee \alpha_2) - \alpha_1, \text{ and } \beta_2 = (\alpha_1 \vee \alpha_2) - \alpha_2.$$

Then the **S-polynomial** of h_1, h_2 is defined as –

$$\mathsf{S}_\prec(h_1, h_2) = c_2\mathbf{y}^{\beta_1}h_1 - c_1\mathbf{y}^{\beta_2}h_2.$$

Observe that, if $\mathsf{in}_\prec(h_2)$ divides $\mathsf{in}_\prec(h_1)$, then $\mathsf{S}_\prec(h_1, h_2)$ is the first step in the reduction of h_1 by h_2.

We will now define S-polynomials over a surjective ring homomorphism ϕ, and refer to it as *Shadow S-polynomial.*

Definition 2.7. Given two polynomials $f, g \in k[x_1, \ldots, x_n]$, a surjective ring homomorphism $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$, and a term order $\prec$ on $k[y_1, \ldots, y_m]$, the **Shadow S-polynomial** is defined as

$$\mathsf{ShadowSpoly}_{\phi,\prec}(f, g) \triangleq h_1 f - h_2 g$$

where $h_1, h_2 \in k[x_1, \ldots, x_n]$ such that

$$\phi(h_1 f - h_2 g) = \mathsf{S}_\prec(\phi(f), \phi(g)).$$

```
Algorithm 2.2: ShadowSpoly($f, g, \phi, \prec$)
    Data:
        • Two polynomials $f, g \in k[x_1, \ldots, x_n]$ such that $\phi(f) \neq 0$ and $\phi(g) \neq 0$
        • a surjective ring homomorphism $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$
        • a term order $\prec$ over $k[y_1, \ldots, y_m]$
        • an oracle that computes any one member of $\phi^{-1}(m)$ for any monomial $m$ of
          $k[y_1, \ldots, y_m]$
    Result: Two polynomials $h_1, h_2 \in k[x_1, \ldots, x_n]$ such that
                $h_1 f - h_2 g = \mathsf{ShadowSpoly}_{\phi,\prec}(f, g)$.
    Let $c_1 y^{\alpha_1} = \mathsf{in}_\prec(\phi(f))$, and $c_2 y^{\alpha_2} = \mathsf{in}_\prec(\phi(g))$ ;
    $\beta_1 \leftarrow (\alpha_1 \vee \alpha_2) - \alpha_1$ ;
    $\beta_2 \leftarrow (\alpha_1 \vee \alpha_2) - \alpha_2$ ;
    return $h_1 \leftarrow \phi^{-1}(c_2 y^{\beta_1})$, and $h_2 \leftarrow \phi^{-1}(c_1 y^{\beta_2})$ ;
```

Algorithm Algorithm 2.2 computes $h_1, h_2 \in k[x_1, \ldots, x_n]$ for any pair of polynomials $f, g \in k[x_1, \ldots, x_n]$ such that $h_1 f - h_2 g = \mathsf{ShadowSpoly}_{\phi,\prec}(f, g)$.

Observation 4. $(h_1, h_2) = \mathit{ShadowSpoly}(f, g, \phi, \prec) \Rightarrow \phi(h_1 f - h_2 g) = \mathsf{S}_\prec(\phi(f), \phi(g))$.

2.4.2 Shadow division

Let $g, g_1, \cdots, g_s$ be polynomials in $k[x_1, \ldots, x_n]$ and $\prec$ be a term order in $k[x_1, \ldots, x_n]$. Then, the polynomial expression

$$g = \sum_i q_i g_i + r$$

is said to be a **standard expression** for g if

- $\mathsf{in}_\prec(q_i g_i) \preceq \mathsf{in}_\prec(g)$, $\forall i$

- No monomial of r is divisible by $\mathsf{in}_{\prec}(g_i)$ for any i. More formally, no monomial of r belongs to the initial ideal $\langle\ \{\ \mathsf{in}_{\prec}(g_i)\ \mid 1 \leq i \leq s\ \}\ \rangle$.

Here, r is called the **remainder** and q_i's are called the **quotients** of the division of g by $\{g_1, \ldots, g_s\}$.

Standard expression generalizes the concept of divisoin of a polynomial by another polynomial to the concept of division of a polynomial by a set of polynomials. The algorithm to perform such a division is well known (CLO07, Chapter 2, Section 3).

Next we define polynomial division over a ring homomorphism ϕ. We will call it *Shadow Division*.

Definition 2.8. Given a polynomial f and a set of polynomials $B = \{g_1, \ldots, g_s\}$ of $k[x_1, \ldots, x_n]$, a surjective ring homomorphism $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$, and a term order $\prec$ in $k[y_1, \ldots, y_m]$, the **Shadow standard expression** of f w.r.t. B is

$$\bar{f}f = \sum_i q_i g_i + r,$$

where

- $\bar{f}, q_1, \ldots, q_s, r \in k[x_1, \ldots, x_n]$.
- $\phi(\bar{f}) =$ constant.
- The following expression is a standard expression of $\phi(f)$ w.r.t. $\phi(B)$

$$\phi(f) = \frac{1}{\phi(\bar{f})}\left(\sum_j \phi(q_j)\phi(f_j) + \phi(r)\right)$$

Here, r is called the **remainder** and q_i's are called the **quotients** of the division of g by $\{g_1, \ldots, g_s\}$.

Algorithm 2.3 computes the Shadow standard expression of f w.r.t. B.

In step 4, $\mathsf{ShadowSpoly}(p, g_i, \phi, \prec)$ is the first reduction step of $\phi(p)$ by $\phi(g_i)$, since $\mathsf{in}_{\prec}(\phi(g_i))$ divides $\mathsf{in}_{\prec}(\phi(p))$. Hence, the leading term of $\phi(p)$ strictly decreases after each pass of the while loop. Combining this with the fact that $\prec$ is a well-ordering,

Algorithm 2.3: Shadow-Division$(f, \{g_1, \ldots, g_s\}, \phi, \prec)$

Data:

- A polynomial $f \in k[x_1, \ldots, x_n]$
- A set $B = \{g_1, \ldots, g_s\} \in k[x_1, \ldots, x_n]$
- A surjective ring homomorphism $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$
- A term order $\prec$ over $k[y_1, \ldots, y_m]$
- An oracle to compute one member of $\phi^{-1}(m)$ for any monomial m of $k[y_1, \ldots, y_m]$.

Result: Polynomials $\bar{f}, q_1, \ldots, q_s, r \in k[x_1, \ldots, x_n]$ such that

$$\bar{f} f = \sum_i q_i f_i + r$$

is a Shadow standard expression of f w.r.t. B.

```
$\bar{f} \leftarrow 1;\ q_1 \leftarrow 0, \ldots, q_s \leftarrow 0;\ r \leftarrow 0;\ p \leftarrow f$ ;
while $\phi(p) \neq 0$ do
    if $\exists i$ such that $\phi(g_i) \neq 0$ and $\mathsf{in}_\prec(\phi(g_i)) \mid \mathsf{in}_\prec(\phi(p))$ then
        $(h_1, h_2) \leftarrow \mathsf{ShadowSpoly}(p, g_i, \phi, \prec)$ ;        // $\phi(h_1)$ is a constant
        $\bar{f} \leftarrow \bar{f} * h_1;\ q_1 \leftarrow q_1 * h_1, \ldots, q_s \leftarrow q_s * h_1;\ r \leftarrow r * h_1$ ;
        $p \leftarrow p * h_1 - g_i * h_2$ ;
        $q_i \leftarrow q_i + h_2$ ;
    else
        $h \leftarrow \phi^{-1}(\mathsf{in}_\prec(\phi(p)))$ ;
        $r \leftarrow r + h;\ p \leftarrow p - h$ ;                                /* $\phi(p) = 0$ */
    end
end
$r \leftarrow r + p$ ;
return $\bar{f}, q_1, \ldots, q_s, r$ ;
```

we observe that the Shadow-Division algorithm terminates. Reduction of $\phi(p)$ by $\phi(g_i)$ also ensures that $\phi(h_1) = \text{constant}$, and consequently

$$\phi(\bar{f}) = \text{constant}$$

One also observes that

$$\bar{f} \cdot f = \sum_j q_j g_j + r + p$$

is an invariant of the *while* loop. Thus we have the following claim –

Lemma 2.9. *Algorithm 2.3, Shadow-Division $(f, B, \phi, \prec)$, terminates to give*

$$\bar{f} \cdot f = \sum_j q_j \cdot g_j + r,$$

and

$$\phi(f) = \left(\frac{1}{\phi(\bar{f})}\right)\left(\sum_j \phi(q_j)\phi(g_j) + \phi(r)\right)$$

is a standard expression for $\phi(f)$ under $\prec$, where $\phi(\bar{f})$ is a non-zero constant.

2.4.3 Shadow Gröbner Basis

We will now present the notion of *Shadow-Gröbner basis*, and an algorithm to compute such a basis.

Definition 2.10. Let $B = \{f_1, \ldots, f_s\} \subseteq k[x_1, \ldots, x_n]$ be a set of polynomials, $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$ be a surjective ring homomorphism, and $\prec$ be a term order in $k[y_1, \ldots, y_m]$. Then a subset $\mathcal{G} \subset k[x_1, \ldots, x_n]$ such that

- $\langle \mathcal{G} \rangle = \langle B \rangle$, and
- $\phi(\mathcal{G})$ is a Gröbner basis of $\phi(\langle B \rangle)$;

is called a **Shadow-Gröbner basis** of the ideal generated by B.

Algorithm 2.4 to computes Shadow-Gröbner basis, which is a modification of the Buchberger's algorithm (Algorithm B.2).

The following claims ensure that the algorithm terminates, and that it correctly computes Shadow-Gröbner basis.

Algorithm 2.4: Shadow-Buchberger($B, \phi, \prec$)

```
Data:
    • $B = \{f_1, \dots, f_s\} \subseteq k[x_1, \dots, x_n]$
    • a surjective ring homomorphism $\phi : k[x_1, \dots, x_n] \to k[y_1, \dots, y_m]$
    • a term order $\prec$ in $k[y_1, \dots, y_m]$
Result: A Shadow-Gröbner basis of $\langle B \rangle$
1  $B_{\text{new}} \leftarrow B$ ;
2  repeat
3      $B_{\text{old}} \leftarrow B_{\text{new}}$ ;
4      for each pair $f_1, f_2 \in B_{new}$ such that $f_1 \neq f_2$ and $\phi(f_1) \neq 0, \phi(f_2) \neq 0$
       do
5          $(g_1, g_2) \leftarrow$ ShadowSpoly($f_1, f_2, \phi, \prec$) ;
6          Compute Shadow-Division($g_1 f_1 - g_2 f_2, B_{\text{new}}, \phi, \prec$) ;
7          if $\phi(r) \neq 0$ then
8              $B_{\text{new}} \leftarrow B_{\text{new}} \bigcup \{r\}$ ;
9          end
10     end
11 until $\phi(B_{new}) = \phi(B_{old})$;
12 $\mathcal{G} \leftarrow B_{\text{new}}$ ;
13 return $\mathcal{G}$ ;
```

Lemma 2.11. *Algorithm 2.4 terminates.*

Proof. The *repeat* loop iterates only if we detect that $\phi(B_{\text{new}}) \neq \phi(B_{\text{old}})$. And this can only happen if a polynomial r gets added to B_{new} in step 8. The polynomial r has the property that it is the remainder of Shadow-Division of $g_1 f_1 - g_2 f_2$ by B_{new}. Thus, from Lemma 2.9, we have

$$\mathsf{in}_{\prec}(\phi(r)) \notin \langle \, \{ \, \mathsf{in}_{\prec}(\phi(g)) \ \mid g \in B_{\text{new}} \, \} \, \rangle.$$

So, in each iteration of the *repeat* loop, as r gets added to B_{new}, the initial ideal of B_{new} in the image space grows. But $k[y_1, \dots, y_m]$ is Noetherian so the ideal cannot grow indefinitely. Hence the *repeat* loop must terminate. ■

Lemma 2.12. $\langle\ B\ \rangle = \langle\ \mathcal{G}\ \rangle$.

Proof. The remainder r is appended to the basis in the successive iterations of the *repeat* loop (step 8). r is the output of the Shadow-Division algorithm (Algorithm 2.3). So, from Lemma 2.9, we have

$$\bar{f}\,(g_1 f_1 - g_2 f_2) = \sum_i q_i f_i + r,$$

where f_i's are in B_{new}. This shows that r is in the ideal generated by the B_{new}. Hence $\langle\ B_{\text{old}}\ \rangle = \langle\ B_{\text{new}}\ \rangle$, i.e., $\langle\ B_{\text{new}}\ \rangle$ remains constant throughout the algorithm. Since initial value of B_{new} is B_{old} and the final value is $\mathcal{G}$, $\langle\ B\ \rangle = \langle\ \mathcal{G}\ \rangle$. ■

Lemma 2.13. *$\mathcal{G}$ is the shadow-Gröbner basis of $\langle\ B\ \rangle$.*

Proof. Upon termination of the *repeat* loop, the set of polynomials B_{new} has the property that the remainder of Shadow-Division(ShadowSpoly($f_1, f_2, \phi, \prec$), $B_{\text{new}}, \phi, \prec$) is zero for every $f_1, f_2 \in B_{\text{new}}$, where Shadow-Division is computed using Algorithm 2.3. This shows that $\phi(\mathcal{G})$ satisfies the *Buchberger's Criterion* (CLO07, Chapter 2, Section 7) and hence $\phi(\mathcal{G})$ is a Gröbner basis of $\langle\ \phi(B)\ \rangle$. This, combined with the fact that $\langle\ \mathcal{G}\ \rangle = \langle\ B\ \rangle$ (Lemma 2.12), we have that $\mathcal{G}$ is the Shadow-Gröbner basis of $\langle\ B\ \rangle$. ■

Here a remark on the time complexity of Shadow-Buchberger algorithm (Algorithm 2.4) is in order. We have assumed that there exists an oracle which gives us one member of $\phi^{-1}(m)$, for any monomial m. If the computation of ϕ and ϕ^{-1} require times proportional to the size of the input then, Shadow-Buchberger algorithm and Buchberger's algorithm have the same time complexity.

2.4.4 Shadow reduced Gröbner basis

In this section, we will define *Shadow reduced Gröbner basis*, and present an algorithm to compute it.

Definition 2.14. Let $B = \{f_1, \ldots, f_s\} \subseteq k[x_1, \ldots, x_n]$ be a set of polynomials, $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$ be a surjective ring homomorphism, and $\prec$ be a term order in $k[y_1, \ldots, y_m]$. A subset $\mathcal{G} \subset k[x_1, \ldots, x_n]$ such that

- $\phi(\mathcal{G})$ is the reduced Gröbner basis of $\phi(\langle\ B\ \rangle)$, and

```
Algorithm 2.5: reduced-Shadow-Gröbner($B, \phi, \prec$)
    Data:
        • $B = \{f_1, \ldots, f_s\} \subseteq k[x_1, \ldots, x_n]$
        • a surjective ring homomorphism $\phi : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m]$
        • a term order $\prec$ in $k[y_1, \ldots, y_m]$
    Result: A set $G \subseteq k[x_1, \ldots, x_n]$, such that it is the reduced
            Shadow-Gröbner basis of $I$.
 1  $G \leftarrow$ Shadow-Buchberger($B, \phi, \prec$) ;
 2  $B_{\text{old}} \leftarrow G$ ;
 3  $B_{\text{new}} \leftarrow G$ ;
 4  for each $f \in B_{old}$ do
 5      Compute Shadow-Division($f, B_{\text{new}} \setminus \{f\}, \phi, \prec$) ;
 6      $B_{\text{new}} \leftarrow B_{\text{new}} \setminus \{f\}$ ;
 7      if $r \neq 0$ then
 8          $B_{\text{new}} \leftarrow B_{\text{new}} \bigcup \{r\}$ ;
 9      end
10  end
11  $G \leftarrow B_{\text{new}}$ ;
12  return $G$ ;
```

- $\langle\ \mathcal{G}\ \rangle \subseteq \langle\ B\ \rangle$.

- for each $f \in \langle\ B\ \rangle$ such that $\phi(f) \neq 0$, there is $h \in \phi^{-1}(1)$ such that $hf \in \langle\ \mathcal{G}\ \rangle$

is called a **reduced Shadow-Gröbner basis** .

Algorithm 2.5 shows how to compute Shadow reduced Gröbner basis of an ideal in $k[x_1, \ldots, x_n]$ from a generating set of the ideal.

Observation 5. *Algorithm 2.5 terminates.*

Proof. The *for* loop in the steps 4 through 10 iterates over the fixed finite set B_{old}, hence the algorithm terminates. ■

Lemma 2.15. *$\phi(G)$ is the reduced Gröbner basis of $\phi(\langle\ B\ \rangle)$.*

Proof. Consider an iteration of the *for* loop in the steps 4 through 10. Let $f \in B_{\text{old}}$ be the member currently being reduced by $B_{\text{new}} \setminus \{f\}$ (step 5). Also, let r be a member added to B_{new} (step 8) and $\mathbf{y}^\alpha$ be any term of $\phi(r)$. Then from Lemma 2.9, we have $\mathbf{y}^\alpha \notin \mathsf{in}_\prec(\phi(B_{\text{new}} \setminus \{f\}))$. So no term of $\phi(r)$ is divisible by the leading terms of $\phi(B_{\text{new}})$. If in an iteration of the *for loop* (steps 4 to 10) f is replaced by r, then $\phi(r)$ is irreducible by $\phi(B_{\text{new}})$. Since the initial ideal of $\langle\ \phi(B_{\text{new}})\ \rangle$ is an invariant of the loop ($\phi(B_{\text{new}})$ is Gröbner Basis), $\phi(r)$ remains irreducible in all subsequent iterations. After the *for loop* terminates, this holds for all the members of B_{new}. Hence $\phi(G)$ is the reduced Gröbner basis of $\langle\ B\ \rangle$. ■

Lemma 2.16. $\langle\ G\ \rangle \subseteq \langle\ B\ \rangle$.

Proof. The initial value of B_{new} is a Shadow-Gröbner basis of $\langle\ B\ \rangle$, and from Lemma 2.12, it is a basis of $\langle\ B\ \rangle$. In the following *for* loop (steps 4 through 10), let $f \in B_{\text{old}}$ be the polynomial that is currently being reduced. We replace $f \in B_{\text{new}}$ by the r, which is the result of shadow reduction of f by $B_{\text{new}} \setminus \{f\}$ (step 5). Thus, Lemma 2.9 implies that the ideal generated by B_{new} before the replacement of f by r contains the ideal generated by B_{new} after the replacement. The final value of B_{new} is G and the initial value is a Shadow-Gröbner basis of $\langle\ B\ \rangle$, so $\langle\ G\ \rangle \subseteq \langle\ B\ \rangle$. ■

Lemma 2.17. *For every $f \in \langle\ B\ \rangle$ such that $\phi(f) \neq 0$, there exists $h \in \phi^{-1}(1)$ such that $hf \in \langle\ G\ \rangle$.*

Proof. Consider an arbitrary iteration of the *for* loop, and let $f \in B_{\text{old}}$ be the polynomial that is currently being reduced, and r is its shadow reduction. Then using the notations from Lemma 2.9, we have

$$\bar{f}f = \sum_i g_i f_i + r,$$

where f_i's belong to $B_{\text{new}} \setminus \{f\}$. Since $\phi(\bar{f}) = c$ (a constant), the ideal generated by B_{new} after the substitution contains hf where $h = \bar{f}/c \in \phi^{-1}(1)$. ■

Combining Lemmas 2.15, 2.16 and 2.17, we have that the output of reduced-Shadow-Gröbner indeed computes reduced Shadow-Gröbner basis.

2.5 Binomial ideals

In this section, we will prove a few useful results about binomials ideals.

Lemma 2.18. *Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal generated by a set of binomials B. Let a binomial $\mathbf{x}^\alpha - \mathbf{x}^\beta$ in I have an expression*

$$\mathbf{x}^\alpha - \mathbf{x}^\beta = c_1 \mathbf{x}^{\delta_1} \left(\mathbf{x}^{\alpha_1} - \mathbf{x}^{\beta_1}\right) + \cdots + c_s \mathbf{x}^{\delta_s} \left(\mathbf{x}^{\alpha_s} - \mathbf{x}^{\beta_s}\right),$$

where $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in B$ for all i. Then, there is an expression for $\mathbf{x}^\alpha - \mathbf{x}^\beta$

$$\mathbf{x}^\alpha - \mathbf{x}^\beta = \mathbf{x}^{\delta_{j_1}} \left(\mathbf{x}^{\alpha_{j_1}} - \mathbf{x}^{\beta_{j_1}}\right) + \cdots + \mathbf{x}^{\delta_{j_t}} \left(\mathbf{x}^{\alpha_{j_t}} - \mathbf{x}^{\beta_{j_t}}\right)$$

where the binomials on the R.H.S. are members of the first expression and

(i) $\mathbf{x}^{\delta_{j_1}} \cdot \mathbf{x}^{\alpha_{j_1}} = \mathbf{x}^\alpha$

(ii) $\mathbf{x}^{\delta_{j_t}} \cdot \mathbf{x}^{\alpha_{j_t}} = \mathbf{x}^\beta$

(iii) $\mathbf{x}^{\delta_{j_i}} \cdot \mathbf{x}^{\beta_{j_i}} = \mathbf{x}^{\delta_{j_{i+1}}} \cdot \mathbf{x}^{\alpha_{j_{i+1}}}$, $1 \leq i < t$.

Such an expression for a binomial will be called a **chain expansion** .

Proof. From the given expression of $\mathbf{x}^\alpha - \mathbf{x}^\beta$, we construct a weighted undirected graph (V, E) as follows. The monomials in the R.H.S. of the expression form the set of vertices V. There is an edge $(\mathbf{x}^\alpha, \mathbf{x}^\beta)$ in the graph if there exists j such that $b_j = \mathbf{x}^\alpha - \mathbf{x}^\beta$. The weight of the edge is the coefficient of b_j. We observe that the sum of the binomials in the graph with edge weights is equal to $\mathbf{x}^\alpha - \mathbf{x}^\beta$.

We claim that this graph is connected. If not, it has atleast two disconnected components. The sum of the coeffiecients of the binomials forming a component is 0. So, either the polynomial due to a component is 0 or has even number of terms. The set of vertices of the components are also disjoint. So, the sum of the components has atleast 4 distinct terms. But this is absurd as the L.H.S. has only two terms. Thus, the graph is connected.

Now that the graph is connected, a path from $\mathbf{x}^\alpha$ to $\mathbf{x}^\beta$ gives a desired chain expansion of $\mathbf{x}^\alpha - \mathbf{x}^\beta$. ■

Lemma 2.19. *Let I be a binomial ideal in $k[x_1, \ldots, x_n]$. If $f \in I$, then $\exists$ binomials $\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i} \in I$ such that*

$$f = \sum_i c_i \left(\mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i}\right), \tag{2.1}$$

and each $\mathbf{x}^{\alpha_i}, \mathbf{x}^{\beta_i}$ is a member of f.

Proof. Consider a monomial $\mathbf{x}^\delta$ in the R.H.S. of equation 2.1 such that $\mathbf{x}^\delta$ is not a term of f. Let the number of occurances of $\mathbf{x}^\delta$ be l. We can reduce the number of occurances of $\mathbf{x}^\delta$ by the following trick. Assume that $b_j = \mathbf{x}^\delta - \mathbf{x}^\gamma$ and $b_k = \mathbf{x}^\delta - \mathbf{x}^{\gamma'}$ be two binomials containing $\mathbf{x}^\delta$. Then

$$\begin{aligned} f &= \sum_i c_i b_i \\ &= \sum_{i \neq j,k} c_i b_i + c_j(b_j - b_k) + (c+d)b_k \\ &= \sum_{i \neq j,k} c_i b_i + c_j(\mathbf{x}^{\gamma'} - \mathbf{x}^\gamma) + (c+d)b_k. \end{aligned}$$

So, we have reduced the number of occurances of $\mathbf{x}^\delta$. We repeat this procedure to remove all occurances of $\mathbf{x}^\delta$ and all occurances of monomials not belonging to f. ■

2.6 Projection Homomorphism

From this section onwards, we shall restrict our consideration of surjective ring homomorphisms to *projection maps*, and as we have discussed earlier, at the end of section 2.2, the maps in these cases will be denoted by π or Π, with suitable indices.

We would also like to state a certain relabeling of the variables in the polynomial ring $k[x_1, \ldots, x_n]$ so that the description of upcoming algorithms and results become more succint and easy to follow. From now on, we shall partition the set of variables $\{x_1, \ldots, x_n\}$ of the ring $k[x_1, \ldots, x_n]$ into two sets, one that are set to 1 by the projection homomorphism, and the other that are not set to 1. Those that are set to 1 will be denoted by v with suitable indices, and those that are left unchanged will be denoted by u, again with suitable indices.

For example, let the projection homomorphism considered be Π_i, as defined in section 2.2. To recall, Π_i is a projection map from $k[x_1, \ldots, x_n] \to k[x_{i+1}, \ldots, x_n]$. Then the variables $x_1, \ldots, x_i$ will be relabelled $v_1, \ldots, v_i$, and the variables $x_{i+1}, \ldots, x_n$ will be relabelled $u_{i+1}, \ldots, u_n$, respectively.

With this notation, we now describe the steps to compute ϕ^{-1} in the algorithms ShadowSpoly (Algorithm 2.2, step 4) and Shadow-Division (Algorithm 2.3, step 9). We will assume that

$$\phi = \Pi_i.$$

So, using the relabelling,

$$\Pi_i : k[v_1, \ldots, v_i, u_{i+1}, \ldots, u_n] \to k[u_{i+1}, \ldots, u_n].$$

In algorithm ShadowSpoly, we have

$$c_1\mathbf{u}^{\alpha_1} = \mathsf{in}_{\prec}\left(\phi(f)\right), \; c_2\mathbf{u}^{\alpha_2} = \mathsf{in}_{\prec}\left(\phi(g)\right), \qquad \text{(step 1)}.$$

This implies that f and g must contain sub-polynomials of the form $p_1(\mathbf{v})\mathbf{u}^{\alpha_1}$ and $p_2(\mathbf{v})\mathbf{u}^{\alpha_2}$ respectively, such that

$$\phi(p_1(\mathbf{v})) = c_1 \quad \phi(p_2(\mathbf{v})) = c_2.$$

Moreover, if $f' = f - p_1(\mathbf{v})\mathbf{u}^{\alpha_1}$, then $\mathsf{in}_{\prec}\left(\phi(f')\right)$ is strictly less than $\mathsf{in}_{\prec}\left(\phi(f)\right)$. Similar is the case for $g - p_2(\mathbf{v})\mathbf{u}^{\alpha_2}$. We define step 4 as

$$h_1 \leftarrow p_2(\mathbf{v})\mathbf{y}^{\beta_1} \text{ and } h_2 \leftarrow p_1(\mathbf{v})\mathbf{y}^{\beta_2}.$$

In algorithm Shadow-Division, there must exist a sub-polynomial $l(\mathbf{v})\mathbf{u}^{\alpha}$ in $p(\mathbf{x})$ such that

$$\mathsf{in}_{\prec}\left(\phi(p - l(\mathbf{v})\mathbf{u}^{\alpha})\right) \prec \mathsf{in}_{\prec}\left(\phi(p)\right)$$

We then define step 9 as

$$h \leftarrow l(\mathbf{v})\mathbf{u}^{\alpha}.$$

Some properties of the Shadow algorithms in the context of projection homomorphisms are as follows.

Observation 6. *If ϕ is a projection homomorphism, f_1, f_2 are homogeneous w.r.t. a grading vector $\vec{d}$ and $(g_1, g_2) =$ ShadowSpoly$(f_1, f_2, \phi, \prec)$, then $g_1 f_1 - g_2 f_2$ is also homogeneous w.r.t. $\vec{d}$.*

If a binomial $f = \mathbf{x}^{\alpha_1} - \mathbf{x}^{\alpha_2}$ is such that $\phi(f)$ is non-zero, then $\phi(\mathbf{x}^{\alpha_1}) \neq \phi(\mathbf{x}^{\alpha_2})$. Thus, in the case of binomials, the polynomials g_1, g_2 in step 4 of algorithm ShadowSpoly, and h in step 9 of Shadow-Buchberger algorithm are monomials.

Observation 7. *Let ϕ be a projection homomorphism, and f_1, f_2 be binomials of $k[x_1, \ldots, x_n]$. Moreover, let $(g_1, g_2) =$ ShadowSpoly$(f_1, f_2, \phi, \prec)$. Then $\phi(f_1 g_1 - f_2 g_2)$ is the Shadow S-polynomial of f_1 and f_2, and $f_1 g_1 - f_2 g_2$ is a binomial.*

Observation 8. *Let ϕ be a projection homomorphism, as before, and B be a set of binomials of $k[x_1, \ldots, x_n]$. Then $\bar{f}$, computed by* Shadow-Division$(f, B, \phi, \prec)$*, is a monomial for $f \in k[x_1, \ldots, x_n]$. Additionally, if f and each member of B is homogeneous, then so is the remainder r.*

We have earlier seen (Lemma 2.9) that $\phi(\bar{f})$ is a non-zero constant, and the above observation states that $\bar{f}$ is a monomial. Then in that case, by using the notation for the variables discussed at the start of the section, we see that $\bar{f}$ computed by Shadow-Division algorithm is of the form $\mathbf{v}^\alpha$, for some $\alpha \in \mathbb{N}^i$. Using this fact in the proof of Lemma 2.17, we get the following lemma which is at the heart of algorithm proposed in the next section.

Lemma 2.20. *If ϕ is a projection homomorphism, B is a set of binomials of $k[x_1, \ldots, x_n]$, and G is computed by reduced-Shadow-Buchberger$(B, \phi, \prec)$, then for each binomial $f \in \langle\ B\ \rangle$ such that $\phi(f) \neq 0$, there exists a monomial $\mathbf{v}^\alpha$ such that, $\mathbf{v}^\alpha f \in \langle\ G\ \rangle$.*

2.7 A fast algorithm for computing toric ideals

We have discussed earlier (section 1.3) that saturation of an ideal with respect to the set of variables in the ring is the most important and time comsuming step in the computation of a toric ideal. Now we present Algorithm 2.6 which takes a set of pure difference binomials B and computes $\langle\ B\ \rangle : (x_1 \cdots x_n)^\infty$.

In the following, the projection map Π_i maps $k[\mathbf{v}, \mathbf{u}, y]$ to $k[\mathbf{u}, y]$ by setting each v_j to 1.

To prove the correctness of the algorithm we will use the following notations. We will index the sets in the end of various iterations of the *for* loop (steps 1 to 6) by the counter i of the *for* loop. For example, in the i^{th} iteration the final value of $\tilde{G}$ will be denoted $\tilde{G}_i$. Therefore initial value of B will be denoted by B_n. We will prove the correctness of Algorithm 2.6 by induction on i.

Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal. A polynomial $x_i^{a_i} f$ will be called **Shadow-Saturated** at level i if

$$x_i^{a_i} \cdots x_n^{a_n} y^b f \in I \implies x_1^{a_1} \cdots x_{i-1}^{a_{i-1}} y^{b'} f \in I,$$

Algorithm 2.6: Computation of $I : (x_1 \cdots x_n)^\infty$ for a binomial ideal I

```
Data: a finite set of binomials $B \subset k[x_1, \dots, x_n]$
Result: A generating set of $\langle\, B \,\rangle : (x_1 \dots x_n)^\infty$
for $i = n - 1$ to $0$ do
    $\vec{d} \leftarrow \left( \underbrace{0, \dots, 0,}_{i \text{ components}} \underbrace{1, \dots, 1,}_{n-i \text{ components}} \underbrace{1}_{\text{homogenizing component}} \right)$ ;
    $\tilde{A} \leftarrow \left\{ \tilde{f} \mid f \in B \right\}$ ;
    $\tilde{G} \leftarrow$ reduced-Shadow-Buchberger$(\tilde{A}, \Pi_i, \prec_{\vec{d},i+1})$ ;
    $B \leftarrow \pi_y(\tilde{G} \div x_{i+1}^\infty)$ ;
end
return $B$ ;
```

for some a_j for $1 \leq j \leq i - 1$ and some b'. Ideal I will be called Shadow-Saturated if the above statement is true for all polynomials in I.

Let $\tilde{G}_i$ be the final value of $\tilde{G}$ in the i^{th} iteration. Also, let $c = u_{i+1}^{a_{i+1}} y^a \left(\mathbf{x}^\alpha - y^b \mathbf{x}^\beta\right)$ be a binomial in $\langle\, \tilde{G}_i \,\rangle$. Then, by Lemma 2.18,

$$c = u_{i+1}^{a_{i+1}} y^a \left(\mathbf{x}^\alpha - y^b \mathbf{x}^\beta\right) = y^{a_1} \mathbf{x}^{\delta_1} b_1 + \dots + y^{a_m} \mathbf{x}^{\delta_m} b_m, \tag{2.2}$$

where $b_j = \left(\mathbf{x}^{\alpha_j} - y^{b_j} \mathbf{x}^{\beta_j}\right) \in \tilde{G}_i$ for all j and R.H.S. is a chain.

Lemma 2.21. *If in the chain expansion of c, we have*

$$\mathsf{in}_{\prec_{\vec{d},i+1}} \left(\Pi_i \left(y^{a_j} \mathbf{x}^{\delta_j} \left(\mathbf{x}^{\alpha_j} - y^{b_j} \mathbf{x}^{\beta_j}\right)\right)\right) \preceq \mathsf{in}_{\prec_{\vec{d},i+1}} \left(\Pi_i \left(c\right)\right) \ \textit{for all } j,$$

i.e., the leading monomial in the image of the R.H.S. is same as the leading monomial of the image of the L.H.S., then c is Shadow-Saturated in $\langle\, \tilde{G}_i \div u_{i+1}^\infty \,\rangle$ with respect to u_{i+1}.

Proof. Recall that $\prec_{\vec{d},i+1}$ is a graded reverse lexicographic term order with u_{i+1} being the least. So $\Pi_i \left(\mathbf{x}^{\gamma_1} y^{k_1}\right) \prec_{\vec{d},i+1} \Pi_i \left(\mathbf{x}^{\gamma_2} y^{k_2}\right)$ implies that if $u_{i+1}^l \mid \mathbf{x}^{\gamma_2} y^{k_2}$ then $u_{i+1}^l \mid \mathbf{x}^{\gamma_1} y^{k_1}$.

It is given that $\mathsf{in}_{\prec_{\vec{d},i+1}} \left(\Pi_i \left(y^{a_j} \mathbf{x}^{\delta_j} \left(\mathbf{x}^{\alpha_j} - y^{b_j} \mathbf{x}^{\beta_j}\right)\right)\right) \preceq \mathsf{in}_{\prec_{\vec{d},i+1}} \left(\Pi_i \left(c\right)\right)$, for all j. So if u_{i+1}^l divides c then, u_{i+1}^l divides all monomials in the chain expansion of c. Thus c is Shadow-Saturated in $\langle\, \tilde{G}_i \div u_{i+1}^\infty \,\rangle$. ■

Lemma 2.22. *In the chain expansion of c, let*

$$\Pi_i\left(\mathbf{x}^{\alpha_j} - y^{b_j}\mathbf{x}^{\beta_j}\right) = 0$$

for some j and $\langle\ \tilde{G}_i\ \rangle$ is Shadow-Saturated at level $i+1$. The binomial $u_{i+1}^{a_{i+1}} y^a\left(\mathbf{x}^{\alpha} y^{b_j}\mathbf{x}^{\beta_j} - y^b\mathbf{x}^{\beta}\mathbf{x}^{\alpha_j}\right)$ has a chain expansion

$$y^{b_j}\mathbf{x}^{\beta_j}\left(y^{a_1}\mathbf{x}^{\delta_1}b_1 + \cdots + y^{a_{j-1}}\mathbf{x}^{\delta_{j-1}}b_{j-1}\right) + \mathbf{x}^{\alpha_j}\left(y^{a_{j+1}}\mathbf{x}^{\delta_{j+1}}b_{j+1} + \cdots + y^{a_m}\mathbf{x}^{\delta_m}b_m\right)$$

If $u_{i+1}^{a_{i+1}} y^a\left(\mathbf{x}^{\alpha} y^{b_j}\mathbf{x}^{\beta_j} - y^b\mathbf{x}^{\beta}\mathbf{x}^{\alpha_j}\right)$ is Shadow-Saturated at level i in $\langle\ \tilde{G}_i \div u_{i+1}^{\infty}\ \rangle$, then so is c.

Proof. Since $\Pi_i\left(\mathbf{x}^{\alpha_j} - y^{b_j}\mathbf{x}^{\beta_j}\right) = 0$, $\mathbf{x}^{\alpha_j} - y^{b_j}\mathbf{x}^{\beta_j} = \mathbf{u}^{r_j}\left(\mathbf{v}^{s_j} - \mathbf{v}^{t_j}\right)$. Further, $\langle\ \tilde{G}_i\ \rangle$ is Shadow-Saturated at level $i+1$, this binomial is equal to $u_{i+1}^{l}\left(v^{s_j} - \mathbf{v}^{t_j}\right)$. Further $\mathbf{x}^{\alpha_j} - y^{b_j}\mathbf{x}^{\beta_j}$ is a member of $\tilde{G}_i$, hence $\mathbf{x}^{\alpha_j} = u_{i+1}^{l}\mathbf{v}^{s_j}$ and $y^{b_j}\mathbf{x}^{\beta_j} = u_{i+1}^{l}\mathbf{v}^{t_j}$. Also, $\left(\mathbf{v}^{s_j} - \mathbf{v}^{t_j}\right) \in \tilde{G}_i \div u_{i+1}^{\infty}$.

As a result $u_{i+1}^{a_{i+1}} y^a\left(\mathbf{x}^{\alpha} y^{b_j}\mathbf{x}^{\beta_j} - y^b\mathbf{x}^{\beta}\mathbf{x}^{\alpha_j}\right) = u_{i+1}^{r_j+a_{j+1}} y^a\left(\mathbf{x}^{\alpha}\mathbf{v}^{t_j} - y^b\mathbf{x}^{\beta}\mathbf{v}^{s_j}\right)$. It is given to be Shadow Saturated at level i in $\langle\ \tilde{G}_i \div u_{i+1}\ \rangle$, so

$$E_1 \triangleq y^a\left(\mathbf{x}^{\alpha}\mathbf{v}^{t_j} - y^b\mathbf{x}^{\beta}\mathbf{v}^{s_j}\right) \in \langle\ \tilde{G}_i \div u_{i+1}\ \rangle$$

So $E_1 + y^{a+b}\mathbf{x}^{\beta}E_2 = y^a\mathbf{v}^{t_j} - y^{a+b}\mathbf{x}^{\beta}\mathbf{v}^{t_j} = \mathbf{v}^{t_j}y^a\left(\mathbf{x}^{\alpha} - y^b\mathbf{x}^{\beta}\right) \in \langle\ \tilde{G}_i \div u_{i+1}^{\infty}\ \rangle$. Therefore, c is also Shadow-Saturated at level i in $\langle\ \tilde{G}_i \div u_{i+1}^{\infty}\ \rangle$. ■

Lemma 2.23. *Let the chain expansion of c does not contain any element from* $\ker \Pi_i$. *Also let $\mathbf{x}^{\alpha}y^{k}$ be the largest monomial (in the image space) in the chain expansion of c such that $\Pi_i(\mathbf{x}^{\alpha}y^{k})$ is strictly greater than* $\mathsf{in}_{\prec_{\vec{d},i+1}}(\Pi_i(c))$. *Also, let the number of occurances of $\mathbf{x}^{\alpha}y^{k}$ in the chain be l. Then there exists a chain expansion of c such that either the largest monomial in the chain is strictly less than $\mathbf{x}^{\alpha}y^{k}$, or the largest monomial is still $\mathbf{x}^{\alpha}y^{k}$ and the number of its occurances is less than l.*

Proof. Let $\mathbf{x}^{\alpha}y^{k}$ belong to a the binomial b_i in the chain expansion of c. Without loss of generality, assume that b_i and b_{i+1} share the monomial $\mathbf{x}^{\alpha}y^{k}$.

Then, $y^{a_i}\mathbf{x}^{\delta_i}b_i + y^{a_{i+1}}\mathbf{x}^{\delta_{i+1}}b_{i+1}$ is the Shadow S-polynomial of b_i, b_{i+1} (with some mulplicative factors). As $\tilde{G}_i$ is a Gröbner basis, so there exists a Shadow standard expression for the S-polynomial. Each of the monomials of the Shadow standard expression is strictly less that $\mathbf{x}^{\alpha}y^{k}$. Let the resulting expression of c be

$$c = y^{a_1}\mathbf{x}^{\delta_1}b_1 + \cdots + E + \cdots + y^{a_m}\mathbf{x}^{\delta_m}b_m,$$

where E is the Shadow Standard expression for the S-polynomial. But the expression on the R.H.S. is not necessarily a chain expansion. In that case, we apply Lemma 2.18 to get the desired chain expansion of c. ■

Corollary 2.24. *c has a chain expansion in which leading term of the image of c is largest element in the image of the chain expansion.*

Theorem 2.25. *$\langle\ \tilde{G}_i \div u_{i+1}^{\infty}\ \rangle$ is Shadow-Saturated at level i.*

Proof. Let $c = u_{i+1}^{a_{i+1}} y^a \left(\mathbf{x}^{\alpha} - y^b \mathbf{x}^{\beta}\right)$ be a binomial in $\langle\ \tilde{G}_i\ \rangle$. We will establish that c is Shadow-Saturated at level i in $\tilde{G}_i \div u_{i+1}^{\infty}$. Let the chain expansion of c be

$$u_{i+1}^{a_{i+1}} y^a \left(\mathbf{x}^{\alpha} - y^b \mathbf{x}^{\beta}\right) = y^{a_1}\mathbf{x}^{\delta_1}\left(\mathbf{x}^{\alpha_1} - y^{b_1}\mathbf{x}^{\beta_1}\right) + \cdots + y^{a_m}\mathbf{x}^{\delta_m}\left(\mathbf{x}^{\alpha_m} - y^{b_m}\mathbf{x}^{\beta_m}\right), \quad (2.3)$$

where $\left(\mathbf{x}^{\alpha_j} - y^{b_j}\mathbf{x}^{\beta_j}\right) \in \tilde{G}_i$ for all j.

We can assume that the chain expansion of c does not contain a kernel element because all the kernel elements can be removed from the chain expansion by repeated application of Lemma 2.22 and we will have a binomial c' such that if c' is Shadow-Saturated, then c is also Shadow-Saturated.

From Corollary 2.24 and Lemma 2.21 c is Shadow-Saturated at level i in $\langle\ \tilde{G}_i \div u_{i+1}^{\infty}\ \rangle$.

So, all the binomials in $\tilde{G}_i \div u_{i+1}^{\infty}$ is Shadow-Saturated. From Lemma 2.19 $\langle\ \tilde{G}_i\ \rangle$ is Shadow-Saturated at level i. ■

We conclude from Theorem 2.25 that ideal $\langle\ B_i\ \rangle$ in Algorithm 2.6 is Shadow-Saturated at level i. Hence the final output, B_0, is saturated with respect to $x_1 \cdots x_n$.

Theorem 2.26. *Algorithm 2.6 correctly computes $\langle B \rangle : (x_1 \cdots x_n)^{\infty}$.*

The advantage of the new algorithm is as follows. In this algorithm, the number of variables in the image space is 1 in the first iteration, 2 in the second iteration, and so on. Symbolically let $t(i)$ denote the time complexity of the Büchberger's algorithm in i variable problem. Then, the cost of the proposed algorithm is $\sum_{i=1}^{n} t(i)$ against the Sturmfels' algorithm's cost $n \cdot t(n)$.

Table 2.1:

Number of	Size of basis		Time taken (in sec.)		Speedup
variables	Initial	Final	Sturmfels	Proposed	
6	2	5	0.0	0.00	-
	4	51	0.001	0.00	-
8	4	186	0.12	0.02	6
	6	597	6.58	0.64	10.3
10	6	729	18.16	0.50	36.3
	8	357	2.68	0.29	9.2
12	6	423	4.04	0.27	14.9
	8	2695	822.12	27.21	30.2
14	10	1035	127.97	4.24	30.1

2.8 Experimental Results

In this section we present the results on performance of the new algorithm and compare it with the existing algorithm by Sturmfels (Stu95). In these experiments we randomly generated binomials and computed $J : (x_1 \dots x_n)^\infty$. The programs were written in C. There are cases where one can ignore certain S-polynomial reduction in the Büchberger algorithm for Gröbner basis computation. There is a significant literature on criteria to select such S-polynomials. We only applied one such criterion, referred as *criterion tail* in Proposition 3.15 of (BSR99) in the implementation of the new algorithm as well as to Sturmfels' algorithm. Since every such criterion can be applied to both algorithms, we believe the performance gains shown here will remain same after the implementations are fully optimized.

Table 2.1 shows performances of the two algorithms. Although only a few cases are shown in the table we ran an extensive experiment and in each and every case the proposed algorithm was faster. Also, as expected the performance ratio improves as the number of variables increase.

Chapter 3

A Saturation Algorithm for Homogeneous Binomial Ideals

3.1 Introduction

Let $k[x_1, \ldots, x_n]$ be a polynomial ring in n variables, and let $I \subset k[x_1, \ldots, x_n]$ be a general homogeneous binomial ideal. In this chapter, we describe a fast algorithm to compute the saturation, $I : (x_1 \cdots x_n)^\infty$. This chapter is based on the work (KM11a; KM11b),

3.1.1 Problem Description

Let

$$b = c\mathbf{x}^\alpha + d\mathbf{x}^\beta$$

be a binomial, and $\vec{d} \in \mathbb{Z}^n_{\geq 0}$ be a vector. b is a said to be **homogeneous** w.r.t. $\vec{d}$, if

$$\vec{d} \cdot \alpha = \vec{d} \cdot \beta.$$

The vector $\vec{d}$ is called the **grading vector**.

We describe a fast algorithm to compute the saturation, $I : (x_1 \cdots x_n)^\infty$, of a homogeneous binomial ideal I.

3.1.2 Our Approach

Definition 3.1. (Eis95) Given a ring R, and a multiplicatively closed subset $U \subset R$ not containing zero, we define the **localization** of R at U, written as $R[U^{-1}]$, to be the set of equivalence classes of pairs (r, u) with $r \in R$ and $u \in U$ with the equivalence relation $(r, u) \sim (r', u')$ if there is an element $v \in U$ such that $v(u'r - ur') = 0$ in R. The equivalence class of (r, u) is denoted by r/u. Addition an multiplication operations are defined on $R[U^{-1}]$ as follows:

$$\frac{r}{u} + \frac{r'}{u'} = \frac{u'r + ur'}{uu'} \text{ and } \frac{r}{u} \times \frac{r'}{u'} = \frac{rr'}{uu'}$$

for $r, r' \in R$, and $u, u' \in U$. It can be seen that under these operations $R[U^{-1}]$ is a ring.

We begin by defining some notations. U_i will denote the multiplicatively closed set $\{x_1^{a_1} \cdots x_i^{a_i} \ : \ a_j \geq 0, 1 \leq j < i\}$. $\prec_i$ will denote a graded reverse lexicographic term order with x_i being the least. The grading vector will become clear from the context. $\varphi_i : k[x_1, \ldots, x_n] \to k[x_1, \ldots, x_n][U_i^{-1}]$ will be the natural localization map $r \mapsto r/1$.

Algorithms 3.1 and 3.2 gives the skeletal structure of two algorithms that compute saturation of homogeneous binomial ideals. Algorithm 3.1 describes the saturation algorithm due to (Stu95, Lemma 12.1) To compute $I : (x_1 \cdots x_n)^\infty$, the algorithm computes n Gröbner bases in n variables. Algorithm 3.2 describes the proposed algorithm. The primary motivation for the new approach is that the time complexity of Gröbner basis is a strong function of the number of variables. In the proposed algorithm, a Gröbner basis is computed in the i-th iteration in $n - i$ variables. To do this, the algorithm requires the computation of a Gröbner basis over the ring $k[x_1, \ldots, x_n][U_i^{-1}]$, for $1 \leq i \leq n$. The Gröbner basis over such rings is not known in the literature. Thus, we propose a generalization of Gröbner basis, called *pseudo Gröbner basis*, and appropriately modify the Buchberger's algorithm to compute it.

Algorithm 3.1: Sturmfels' Algorithm

```
Data: A homogeneous binomial ideal, $I \subset k[x_1, \dots, x_n]$.
Result: $I : (x_1 \cdots x_n)^\infty$
for $i \leftarrow 1$ to $n$ do
    $G \leftarrow$ Gröbner basis of $I$ w.r.t. $\prec_i$ ;
    $I \leftarrow \langle\{\ f \div x_i^\infty \mid f \in G\ \}\rangle$ ;
end
return $I$ ;
```

Algorithm 3.2: Proposed Algorithm

```
Data: A homogeneous binomial ideal, $I \subset k[x_1, \dots, x_n]$.
Result: $I : (x_1 \cdots x_n)^\infty$
for $i \leftarrow n$ to $1$ do
    $G \leftarrow$ Pseudo Gröbner basis of $\varphi_i(I)$ w.r.t. $\prec_i$ ;
    $I \leftarrow \langle\ \{\ \varphi_i^{-1}(f : x_i^\infty) \mid f \in G\ \}\ \rangle$ ;
end
return $I$ ;
```

3.1.3 Refined Problem Statement

Let R be a commutative Noetherian ring with unity, and $U \subset R$ be a multiplicatively closed set with unity but without zero. Let the set U^+ be defined as

$$U^+ = \{u \ : \ u \in U, \ \text{or} \ -u \in U, \ \text{or} \ u = 0\}.$$

Let S denote the localization of R w.r.t U, i.e., $S = R[U^{-1}]$. Define a class of binomials, called U-binomials, in the ring $S[x_1, \ldots, x_n]$ as follows

$$\frac{u_1}{u'_1}\mathbf{x}^{\alpha_1} + \frac{u_2}{u'_2}\mathbf{x}^{\alpha_2},$$

where $u_i \in U^+, u'_i \in U$.

We will address the problem of efficiently saturating a homogeneous U-binomial ideal w.r.t. all the variables in the ring, namely $x_1, \ldots, x_n$.

This problem is a generalization of some well studied problems. If R is a field, then this problem reduces to saturating a binomial ideal in the standard polynomial ring. The class of problems we have by restricting R to a field and U to $\{+1, -1\}$ includes the problem of saturation of pure difference binomial ideals and computation of toric ideals.

The rest of the chapter is arranged as follows. Sections 2 and 3 deal with "chain binomials" and "chain sums" for general binomial ideals. Section 4 deals with reductions of a U-binomial by a set of U-binomials. In section 5, we will present the notion of pseudo Gröbner Basis for $S[x_1, \ldots, x_n]$, and a modified Buchberger's algorithm to compute it. In section 6, we present a result similar to Sturmfels' lemma (Stu95, Lemma 12.1). The final saturation algorithm is presented in section 7. Finally, in section 8, we present some preliminary experimental results comparing our algorithm applied to toric ideals, to that of Sturmfels' algorithm and Project and Lift algorithm (HM09).

3.2 Chain and chain-binomial

In this section we shall describe the terminology we will need to work with general binomials in the ring $S[x_1, \ldots, x_n]$. Since this polynomial ring is over a ring, $S =$

$R[U^{-1}]$, rather than over a field k, we will revisit the terminology used for general polynomial rings and restate them in the context of $S[x_1, \ldots, x_n]$.

Symbols $u, v, w, \ldots$ will denote elements of U^+ and $u', v', w', \ldots$ will denote the elements of U. A **term** in the polynomial ring $S[x_1, \ldots, x_n]$ is the product of an S element with a monomial, for example, $(r/u')x_1^{a_1} \ldots x_n^{a_n}$ where $r \in R$ and $u' \in U$. To simplify the notations, we may also write it as $(r/u')\mathbf{x}^\alpha$, where α represents the vector $(a_1, \ldots, a_n)$. If $r \in U^+$, then we will call it a U-**term** . A **binomial** is a polynomial with at most two terms, i.e.,

$$b = \frac{r_1}{u'_1}\mathbf{x}^\alpha + \frac{r_2}{u'_2}\mathbf{x}^\beta.$$

If both the terms of a binomial are U-terms, then we will call it a U-**binomial** . A U-binomial of the form

$$\frac{u_1}{u'_1}\mathbf{x}^\alpha + \frac{u_2}{u'_2}\mathbf{x}^\alpha$$

will be called **balanced** . Since U is not necessarily closed under addition, a balanced U-binomial $((u_1/u'_1) + (u_2/u'_2))\mathbf{x}^\alpha$ need not be a U-term in general. A binomial b is said to be **oriented** if one of its terms is identified as *first* (and the other *second*). If b is oriented, then b^{rev} denotes the same binomial with the opposite orientation.

In the above notations, one of the coefficients of a binomial or U-binomial may be zero. Hence, the definition of binomials (rep. U-binomials) includes single terms (resp. U-terms). To be able to handle all binomials in a uniform manner, we shall denote a single term $(r/u')\mathbf{x}^\alpha$ as

$$(r/u')\mathbf{x}^\alpha + (0/1)\mathbf{x}^\square,$$

where $\mathbf{x}^\square$ is a symbolic monomial. This will help in avoiding to consider a separate case for single terms in some proofs. We shall refer to such binomials as **mono-binomials** . In a term-ordering, $\mathbf{x}^\square$ will be defined to be the least element. Coefficient of $\mathbf{x}^\square$ in every occurrence will be zero.

Definition 3.2. A sequence of oriented binomials

$$\frac{r_1}{u'_1}\mathbf{x}^{\beta_1}b_1,\ \frac{r_2}{u'_2}\mathbf{x}^{\beta_2}b_2,\ \ldots\ ,\ \frac{r_q}{u'_q}\mathbf{x}^{\beta_q}b_q$$

possibly with repetitions will be called a **chain** if the second term of $(r_i/u'_i)\mathbf{x}^{\beta_i}b_i$ cancels the first term of $(r_{i+1}/u'_{i+1})\mathbf{x}^{\beta_{i+1}}b_{i+1}$, for each $1 \leq i < q$. Let B be a set of U-binomials. If each b_i in the chain belongs to B, then we will call the chain a B-**chain** . The sum of the binomials of the chain (respectively, B-chain)

$$\tilde{b} = \sum_{i=1}^{q} \frac{r_i}{u'_i} \mathbf{x}^{\beta_i} b_i,$$

which is itself a binomial, will be called the corresponding **chain binomial** (respectively, B-**chain binomial**). It is the first term of $(r_1/u'_1)\mathbf{x}^{\beta_1} b_1$ plus the second term of $(r_q/u'_q)\mathbf{x}^{\beta_q} b_q$, because all the intermediate terms get canceled. We will call any two chains **equivalent** if their corresponding chain-binomials are the same.

In the later sections, we will be interested in the "shape" of a chain. Given a term ordering we will call a chain **ascending** if the first monomial is strictly less than the second monomial in each binomial of the chain with respect to the given term-order. Similarly, **descending chains** chains are also defined. Another shape of significant interest is the one in which there are three sections in the chain: first is descending, second is horizontal (all binomials in it are balanced), and the final section is ascending. Any of these sections can be of length zero. Such chains will be called **bitonic** .

Suppose we have a sequence of oriented U-binomials such that the monomial of the second term of the i-th binomial in the sequence is equal to the monomial of first term of the $(i+1)$-st binomial in the sequence. Then we can multiply suitable coefficients to these U-binomials to turn this sequence into a chain such that its chain-binomial is also a U-binomial. Let

$$\mathbf{x}^{\beta_1} b_1,\ \mathbf{x}^{\beta_2} b_2,\ \ldots,\ \mathbf{x}^{\beta_q} b_q$$

be a sequence of oriented U-binomials such that the first $q-1$ binomials are not mono-binomials. Let

$$\mathbf{x}^{\beta_i} b_i = \mathbf{x}^{\beta_i} \left(\frac{u_i}{u'_i} \mathbf{x}^{\alpha_{i,1}} + \frac{v_i}{v'_i} \mathbf{x}^{\alpha_{i,2}} \right),$$

where $(u_i/u'_i)\mathbf{x}^{\alpha_{i1}}$ is the first term for each i. Let $\beta_i + \alpha_{i,2} = \beta_{i+1} + \alpha_{(i+1),1}$ for all $1 \leq i < q$. Consider the sequence $(\ldots, (d_i/d'_i)\mathbf{x}^{\beta_i} b_i, \ldots), 1 \leq i \leq q$ where $d_1/d'_1 = 1/1$ and

$$\frac{d_i}{d'_i} = (-1)^{i-1} \frac{v_1}{v'_1} \frac{u'_2}{u_2} \frac{v_2}{v'_2} \frac{u'_3}{u_3} \frac{v_3}{v'_3} \cdots \frac{v_{i-1}}{v'_{i-1}} \frac{u'_i}{u_i},$$

for $i > 1$. It is easy to see that it is a chain of U-binomial and its chain-binomial is the U-binomial

$$\frac{u_1}{u_1'}\mathbf{x}^{\alpha_{1,1}} + \frac{d_q}{d_q'}\frac{v_q}{v_q'}\mathbf{x}^{\alpha_{q,2}}$$

which will be denoted by $\mathtt{B}(\mathbf{x}^{\beta_1}b_1, \ldots, \mathbf{x}^{\beta_q}b_q)$. Note that if b_q is a mono-binomial, then the second term will be $(0/1)\mathbf{x}^{\square}$.

Observation 9. *Let $(\mathbf{x}^{\beta_1}b_1, \ldots, \mathbf{x}^{\beta_k}b_k)$ be a sequence of oriented U-binomials where $b_i \in B$ and none of which are mono-binomials. Furthermore, the second monomial of $\mathbf{x}^{\beta_i}b_i$ and the first monomial of $\mathbf{x}^{\beta_{i+1}}b_{i+1}$ are same for all $1 \le i < k$. Then $\mathtt{B}(\mathbf{x}^{\beta_1}b_1, \ldots, \mathbf{x}^{\beta_k}b_k, \mathbf{x}^{\beta_k}b_k^{\mathsf{rev}}, \ldots, \mathbf{x}^{\beta_1}b_1^{\mathsf{rev}}) = 0$.*

3.3 Decomposition into chains

If B is a finite set of pure difference binomials, then every binomial in $\langle B\rangle$ is a B-chain binomial (Lemma 2.18). This property is used in the computation of a toric ideal. In case B has general binomials this property does not hold. But in the following theorem, we will show that in ideals generated by U-binomials every polynomial can be expressed as the sum of some B-chain binomials. This result is used in the proof of theorems 3.10 and 3.11. For any polynomial f, $\mathsf{mon}(f)$ will denote the set of monomials in f.

Theorem 3.3. *Let B be a finite set of U-binomials in $S[x_1, \ldots, x_n]$. For every polynomial f in $I = \langle B\rangle$, there exists a set of B-chain binomials $\tilde{b}_i$ such that $f = \sum_i \tilde{b}_i$ where both monomials of every $\tilde{b}_i$ belongs to $\mathsf{mon}(f)\bigcup\{\mathbf{x}^{\square}\}$.*

Proof. Let $B = \{b_1, \ldots, b_n\}$. Consider an arbitrary polynomial $f \in \langle B\rangle$. So

$$f = \sum_i \frac{r_i}{w_i'}\mathbf{x}^{\beta_i}b_{j_i},$$

where $(r_i/w_i')\mathbf{x}^{\beta_i} \in S[x_1, \ldots, x_n]$, for all i. Define an edge-weighted graph G (multi-edges and loops allowed) representing this expression in the following manner. The vertex set of this graph is the set of distinct monomials in $(r_i/w_i')\mathbf{x}^{\beta_i}b_i$, for all i. Vertices corresponding to $\mathsf{mon}(f)\bigcup\{\mathbf{x}^{\square}\}$ will be called *terminal vertices.*

There is one edge for each binomial in the sum-expression for f. The i-th edge is incident upon the two monomials associated with $\mathbf{x}^{\beta_i} b_i$, if they are distinct. Otherwise it forms a loop on that monomial. Weights are assigned to two halves of each edge separately. Suppose $b_i = (u_i/u'_i)\mathbf{x}^{\alpha_{i,1}} + (v_i/v'_i)\mathbf{x}^{\alpha_{i,2}}$. Then we associate weight $(r_i/w'_i)(u_i/u'_i)$ to the end incident on $\mathbf{x}^{\beta_i}\mathbf{x}^{\alpha_{i,1}}$ and weight $(r_i/w'_i)(v_i/v'_i)$ to the end incident on $\mathbf{x}^{\beta_i}\mathbf{x}^{\alpha_{i,2}}$.

It should be clear from the construction that the sum of end-weights incident upon a non-terminal vertex must be zero. Hence the degree of non-terminal vertices can never be one. Each end-weight incident on $\mathbf{x}^{\square}$ is zero, so their sum is also zero. See example in figure 3.1.

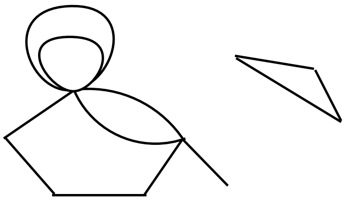

Figure 3.1: An example of chain decomposition graph

Consider any connected component, C, of G. The polynomial corresponding to C is the sum of its monomials, weighted with the sum of end-weights incident on it. This is also the sum of binomials corresponding to the edges in C. So the polynomial associated with G is the sum of polynomials of all components of G, which is f.

If a component does not contain any $\mathsf{mon}(f)$ vertex, then the corresponding polynomial will be zero. So we can delete it from the graph without affecting the total polynomial. Similarly any isolated $\mathsf{mon}(f)$ vertex with no loop-edge also contributes zero and can be deleted from the graph. So we can assume that every connected component of G has at least one $\mathsf{mon}(f)$ vertex and degree of all terminal vertices is at least 1 and as observed earlier, the degree of non-terminal vertices is at least 2.

We will establish the claim of the theorem by induction on the number of edges in the graph. If the graph has one edge, then the corresponding expression is a trivial B-chain binomial with both monomials from $\mathsf{mon}(f) \bigcup \{\mathbf{x}^{\square}\}$. Next we will consider the graphs with more than one edge.

If there is a component with at least two terminal vertices, then select a shortest path w between two different vertices of $\mathsf{mon}(f) \bigcup \{\mathbf{x}^{\square}\}$ in the component. In case all components have only one $\mathsf{mon}(f)$ node, then from lemma 3.4 given below, we conclude that a closed walk w exists passing through the terminal vertex and has at least one edge on it which is traversed only once.

In either case, the walk w has at least one edge on it which is not traversed more than once and both its end-vertices (the two end-vertices may be same if w is a closed walk) are terminals. Furthermore, if one of the end-vertices is $\mathbf{x}^{\square}$, then w must be a path, not a closed walk. Hence, all edges on it are traversed only once. In particular, the edge incident on $\mathbf{x}^{\square}$ is traversed only once.

Let

$$\left(\frac{r_{j_1}}{w'_{j_1}}\mathbf{x}^{\beta_{j_1}}b_{j_1},\ \frac{r_{j_2}}{w'_{j_2}}\mathbf{x}^{\beta_{j_2}}b_{j_2},\ \ldots\ ,\ \frac{r_{j_k}}{w'_{j_k}}\mathbf{x}^{\beta_{j_k}}b_{j_k}\right)$$

be the sequence of the binomials associated with the successive the edges of the walk. Orient these binomials such that walk proceeds from the first to the second term of each binomial. Then the second monomial of i-th binomial is same as the first monomial of $(i+1)$-st binomial of the walk/sequence.

Suppose the t-th edge in the walk is traversed only once. In case the walk ends in $\mathbf{x}^{\square}$, take t to be the edge incident on $\mathbf{x}^{\square}$. Let the t^{th} edge of the walk has index l, i.e. $j_t = l$. Consider the chain binomial

$$\tilde{b} = \mathsf{B}\left(\frac{r_l}{w'_l}\frac{d'_t}{d_t}\mathbf{x}^{\beta_{j_1}}b_{j_1},\ \ldots\ ,\ \frac{r_l}{w'_l}\frac{d'_t}{d_t}\mathbf{x}^{\beta_{j_k}}b_{j_k}\right),$$

where d_t/d'_t is as defined in the end of section 3.2. Observe that in the chain expression of $\tilde{b}$ the t-th binomial is $(r_l/w'_l)\mathbf{x}^{\beta_l}b_l$ and all the remaining binomials correspond to other than l-th edge of the graph. From the definition of binomial $\tilde{b}$, both its monomials are from the set $\{\mathsf{mon}(f)\bigcup\{\mathbf{x}^{\square}\}$.

Let $f' = f - \tilde{b}$. Express f' as a sum expression by combining the sum expressions of f and $\tilde{b}$. The coefficients of a given binomial in the sum expression of f and of $\tilde{b}$ combine to a single coefficient of the form r/u'. Hence, we get a sum-expression for f' where the binomials are the same as in the expression of f but their coefficients may change. The coefficient of $\mathbf{x}^{\beta_l}b_l$ in f' sum-expression is zero. So the number of addend binomials in f' expression is at least one less that that in f expression. Therefore the graph corresponding to f' will have at least one fewer edge then in the graph of f. This establishes the induction-step and hence the proof is complete. ■

Following is a graph theoretic result which was used in the above theorem.

Lemma 3.4. *Let H be an undirected connected graph (possibly with loops and multi-edges) with n vertices. Let s be a specified vertex. Also, let the degree of every vertex other than s be greater than one and $\mathsf{deg}(s) \geq 1$ (so if $n = 1$ then s has a loop). Then, there exists a closed walk passing through s which has at least one edge that occurs only once in it.*

Proof. The number of edges in H is half of the sum of degrees of its vertices, so it is at least $\lceil (1 + 2(n-1))/2 \rceil = n$. A tree on n vertices has $n-1$ edges. So there must exist a cycle in H. Since the graph allows loops and parallel edges, the cycles in the graph include 1-cycles (loop) and 2-cycles (due to parallel edges).

Suppose this cycle is $v_0 \overset{e'_0}{\to} v_1 \overset{e'_1}{\to} \ldots\ v_{m-1} \overset{e'_{m-1}}{\to} v_0, m \geq 1$. Furthermore, suppose v_i is one of the nearest vertices of the cycle from s and let $e_1, e_2, \ldots, e_t$ be a shortest paths from s to v_i. So this path only touches the cycle at v_i and the sets of the edges of the path and the cycle are disjoint. Then the desired walk is $e_1, e_2, \ldots, e_t, e'_i, e'_{i+1}, \ldots, e'_0, e'_1 \ldots, e'_{i-1}, e_t, e_{t-1}, \ldots, e_1$. ■

3.4 Reduction of U-binomials

Let B be a finite set of non-balanced U-binomials (which may include mono-binomials) and a term order $\prec$. In this section, we will formally describe the reduction of any U-binomial by B with respect to the given term order. We will assume that each binomial of B is oriented by setting the leading term as the first term. We will denote the leading term of a binomial b by $in_\prec(b)$.

Given an arbitrary U-term $(u/u')\mathbf{x}^\alpha$, Algorithm 3.3 computes a descending B-chain

$$\frac{v_1}{v'_1}\mathbf{x}^{\beta_1} b_{j_1},\ \ldots\ ,\ \frac{v_p}{v'_p}\mathbf{x}^{\beta_p} b_{j_p}$$

with corresponding B-chain binomial

$$\sum_{i=1}^{p} \frac{v_i}{v'_i}\mathbf{x}^{\beta_i} b_{j_i} = \frac{u}{u'}\mathbf{x}^\alpha - \frac{w}{w'}\mathbf{x}^\gamma,$$

where $\mathbf{x}^\gamma$ is not divisible by the leading term of any member of B. The term $(w/w')\mathbf{x}^\gamma$ will be denoted by $\overline{(u/u')\mathbf{x}^\alpha}^B$.

Algorithm 3.3: Division algorithm for a U-monomial by a set of non-balanced U-binomials

```
Data:
    • A finite set, $B$, of non-balanced $U$-binomials
    • A $U$-term $(u/u')\mathbf{x}^\alpha$
Result: A $U$-term $(w/w')\mathbf{x}^\gamma$ which is irreducible by $B$ and a $B$-chain
        corresponding to binomial $(u/u')\mathbf{x}^\alpha - (w/w')\mathbf{x}^\gamma$.
$(w/w')\mathbf{x}^\gamma \leftarrow (u/u')\mathbf{x}^\alpha$ ;
$i \leftarrow 0$ ;
while some leading monomial in $B$ divides $\mathbf{x}^\gamma$ do
    Select $b = (\mu_1/\mu_1')\mathbf{x}^{\delta_1} + (\mu_2/\mu_2')\mathbf{x}^{\delta_2}$ from $B$ such that the leading
    monomial $\mathbf{x}^{\delta_1}$ divides $\mathbf{x}^\gamma$ ;
    $i \leftarrow i + 1$ ;
    $\mathbf{x}^{\beta_{j_i}} \leftarrow \mathbf{x}^\gamma/\mathbf{x}^{\delta_1}$ ;
    $v_i/v_i' \leftarrow (-w/w')(\mu_1'/\mu_1)$ ;
    $w/w' \leftarrow (v_i/v_i')(\mu_2/\mu_2')$ ;
    $\mathbf{x}^\gamma \leftarrow \mathbf{x}^{\beta_{j_i}} \cdot \mathbf{x}^{\delta_2}$ ;
end
return $(w/w')\mathbf{x}^\gamma$, $\big((v_1/v_1')\mathbf{x}^\beta b_{j_1}, \ \dots\ , \ (v_i/v_i')\mathbf{x}^{\beta_i} b_{j_i}\big)$ ;
```

Any initial sub-chain

$$\frac{v_1}{v_1'}\mathbf{x}^{\beta_1} b_{j_1}, \ \dots\ , \ \frac{v_p}{v_p'}\mathbf{x}^{\beta_p} b_{j_l}$$

is called a **reduction** of $(u/u')\mathbf{x}^\alpha$ and if the corresponding chain-binomial is

$$\frac{u}{u'}\mathbf{x}^\alpha - \frac{w_1}{w_1'}\mathbf{x}^{\gamma_1},$$

then $(u/u')\mathbf{x}^\alpha$ is said to have B-**reduced** to $(w_1/w_1')\mathbf{x}^{\gamma_1}$. In particular, $\overline{(u/u')\mathbf{x}^\alpha}^B$ is the irreducible B-reduction of $(u/u')\mathbf{x}^\alpha$. If $p = \sum_i (u_i/u_i')\mathbf{x}^{\alpha_i}$ and $(w_i/w_i')\mathbf{x}^{\gamma_i}$ be some B-reduction of $(u_i/u_i')\mathbf{x}^{\alpha_i}$ for each i, then $\sum_i (w_i/w_i')\mathbf{x}^{\gamma_i}$ is a B-reduction of p.

The reduction of binomials is of special interest here. Suppose we have a non-balanced U-binomial

$$b = \frac{u_1}{u_1'}\mathbf{x}^{\alpha_1} + \frac{u_2}{u_2'}\mathbf{x}^{\alpha_2}$$

and a finite set B of non-balanced U-binomials in which the first term is greater than the second term. Let $(w_1/w_1')\mathbf{x}^{\gamma_1}$ and $(w_2/w_2')\mathbf{x}^{\gamma_2}$ be the reductions of $(u_1/u_1')\mathbf{x}^{\alpha_1}$ and $(u_2/u_2')\mathbf{x}^{\alpha_2}$ respectively. So

$$b' = \frac{w_1}{w_1'}\mathbf{x}^{\gamma_1} + \frac{w_2}{w_2'}\mathbf{x}^{\gamma_2}$$

is a reduction of b. Adjoining the reduction chain of $(u_1/u_1')\mathbf{x}^{\alpha_1}$ with b' (if it is non-zero) followed by the reverse of the reduction chain of $(u_2/u_2')\mathbf{x}^{\alpha_2}$ results into a bitonic chain called a **reduction chain** of b with respect to B. Obviously, its chain-binomial is b.

In case b is a balanced U-binomial

$$\frac{u_1}{u_1'}\mathbf{x}^{\alpha} + \frac{u_2}{u_2'}\mathbf{x}^{\alpha},$$

we only need to reduce $\mathbf{x}^{\alpha}$. Let a reduction chain and the reduction monomial be C_1 and $(w_1/w_1')\mathbf{x}^{\gamma}$ respectively. Then

$$b' = \frac{u_1}{u_1'}\frac{w_1}{w_1'}\mathbf{x}^{\gamma} + \frac{u_2}{u_2'}\frac{w_1}{w_1'}\mathbf{x}^{\gamma}$$

is a B-reduction of b and the corresponding reduction chain is

$$\frac{u_1}{u_1'}C_1,\ b',\ \frac{u_2}{u_2'}C_1^{\mathsf{rev}}.$$

For any binomial b, any B-reduction chain which reduces it to b', is a $B\bigcup\{b'\}$-chain and it is bitonic. In particular, if b' is zero then the reduction chain will be a B-chain.

Lemma 3.5. *Let C be a B-chain and $b \in B$. Let $B' = B \setminus \{b\}$ and b' be some B'-reduction of b. Then there is a $B'\bigcup\{b'\}$-chain which is equivalent to C.*

Proof. If b does not occur in C, then C is also a $B'\bigcup\{b'\}$-chain.

The reduction chain of b by B' is a $B'\bigcup\{b'\}$-chain. In case b occurs in C, plug this reduction chain in places of b in C. So the resulting chain is equivalent to C and itself a $B'\bigcup\{b'\}$-chain. ■

3.5 Pseudo-Gröbner Basis

In section 3.1.2, we saw that the saturation of an ideal in $k[x_1, \dots, x_n]$ can be computed by first computing a suitable Gröbner basis for it, as described in Sturmfels' lemma (Stu95, Lemma 12.1). Unfortunately, Gröbner basis is only defined for ideals in $k[x_1, \dots, x_n]$, where k is a field, not for $S[x_1, \dots, x_n]$ as is the case here. In this section, we will describe a type of basis for U-binomial ideals in $S[x_1, \dots, x_n]$ which closely resembles a Gröbner basis. In section 3.6, we will also prove a theorem similar to the Sturmfels' lemma which will allows us to compute the saturation of such ideals.

Definition 3.6. For every finite U-binomial set G, G_1 and G_2 will denote its partition, where the former will represent the set of non-balanced binomials and the latter will represent the set of balanced binomials of G.

Definition 3.7. Let

$$b_1 = \frac{u_1}{u_1'}\mathbf{x}^{\alpha_1} + \frac{v_1}{v_1'}\mathbf{x}^{\beta_1},$$
$$b_2 = \frac{u_2}{u_2'}\mathbf{x}^{\alpha_2} + \frac{v_2}{v_2'}\mathbf{x}^{\beta_2}$$

be non-balanced U-binomials belonging to $S[x_1, \dots, x_n]$. Let $\prec$ be a term order and $\mathbf{x}^{\beta_i} \prec \mathbf{x}^{\alpha_i}$ for $i = 1, 2$. Further, let

$$b_3 = \left(\frac{w_1}{w_1'} + \frac{w_2}{w_2'}\right)\mathbf{x}^{\alpha}.$$

We define two types of S-**binomials** as follows: First one for a pair of two non-balanced binomials –

$$\mathsf{S}_{\prec}(b_1, b_2) \triangleq \frac{u_1 v_2}{u_1' v_2'}\mathbf{x}^{\beta_2+\gamma-\alpha_2} - \frac{v_1 u_2}{v_1' u_2'}\mathbf{x}^{\beta_1+\gamma-\alpha_1},$$

where $\mathbf{x}^{\gamma}$ is the LCM of $\mathbf{x}^{\alpha_1}$ and $\mathbf{x}^{\alpha_2}$. The second type is for a balanced and non-balanced binomial. In this case –

$$\mathsf{S}_{\prec}(b_3, b_1) \triangleq \left(\frac{w_1}{w_1'} + \frac{w_2}{w_2'}\right)\mathbf{x}^{\beta_1+\gamma-\alpha_1},$$

where $\mathbf{x}^{\gamma}$ is the LCM of $\mathbf{x}^{\alpha}$ and $\mathbf{x}^{\alpha_1}$. Observe that if b' and b'' are both U-binomials, then $\mathsf{S}_{\prec}(b', b'')$ is also a U-binomial in both cases of S-binomial definition.

Assume a fixed term-order. In a chain

$$\left(\dots\, , \ \frac{v_i}{v'_i}\mathbf{x}^{\beta_i} b_i, \ \dots \right),$$

two consecutive binomials will be said to form a **peak** if at least one is non-balanced and the monomial at their junction is greater than or equal to the other two monomials. Further suppose $\mathbf{x}^{\beta_{i-1}} b_{i-1}$ and $\mathbf{x}^{\beta_{i+j}} b_{i+j}$ are non-balanced binomials and all the intermediate binomials are balanced, then the binomials $\mathbf{x}^{\beta_k} b_k$, $i \le k \le i+j-1$ are called **plateau** if at least one of $(i-1)$-st and i-th binomials or $(i+j-1)$-th and $(i+j)$-th binomials form a peak. See figure 3.2.

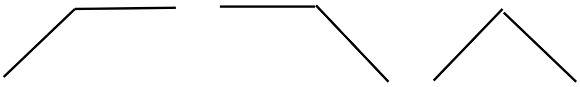

Figure 3.2: Types of peaks

Suppose

$$C = \left(\dots\, , \ \frac{u_{i-1}}{u'_{i-1}}\mathbf{x}^{\beta_{i-1}} b_{i-1}, \ \frac{u_i}{u'_i}\mathbf{x}^{\beta_i} b_i, \ \dots \right)$$

is a chain where $(i-1)$-st and i-th binomials form a peak. In case b_{i-1} and b_i both are non-balanced, then there exists a term $(w/w')\mathbf{x}^{\gamma}$ such that following chain is equivalent to C:

$$\left(\dots\, , \ \frac{u_{i-2}}{u'_{i-2}}\mathbf{x}^{\beta_{i-2}} b_{i-2}, \ \frac{w}{w'}\mathbf{x}^{\gamma} \mathsf{S}_{\prec}\left(b_{i-1}, b_i\right), \ \frac{u_{i+1}}{u'_{i+1}}\mathbf{x}^{\beta_{i+1}} b_{i+1}, \ \dots \right).$$

In the second case, when b_{i-1} is balanced and b_i is non-balanced, then there exists a constant w_1/w'_1 and a term $(w_2/w'_2)\mathbf{x}^{\gamma}$ such that the following chain is equivalent to C:

$$\left(\dots\, , \ \frac{u_{i-2}}{u'_{i-2}}\mathbf{x}^{\beta_{i-2}} b_{i-2}, \ \frac{w_1}{w'_1}\mathbf{x}^{\beta_i} b_i, \ \frac{w_2}{w'_2}\mathbf{x}^{\gamma} \mathsf{S}_{\prec}\left(b_{i-1}, b_i\right), \ \frac{u_{i+1}}{u'_{i+1}}\mathbf{x}^{\beta_{i+1}} b_{i+1}, \ \dots \right).$$

The third case where b_{i-1} is non-balanced and b_i is balanced, is same as the second case with initial chain reversed. Observe that in these cases the original peak is removed, see figure 3.3.

Lemma 3.8. *Let G be a finite set of U-binomials and assume a fixed term-ordering $\prec$. If for every S-polynomial $\mathsf{S}_{\prec}(b_1, b_2)$, where $b_1, b_2 \in G$, has a G-chain in which each monomial is less than or equal to at least one monomial of $\mathsf{S}_{\prec}(b_1, b_2)$, then every G-chain has an equivalent bitonic G-chain.*

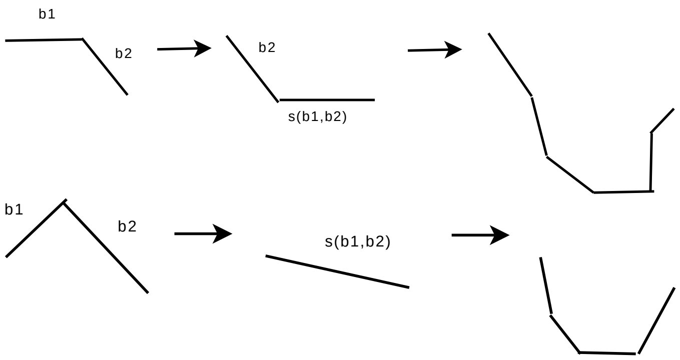

Figure 3.3: S-polynomial reductions

Proof. Consider any arbitrary G-chain. If it has no peak, then it must be bitonic. Otherwise locate one of the highest (in terms of the ordering) peaks. Replace the two binomials forming the peak by the S-polynomial or the combination of the S-polynomial and the non-balanced binomial as described in the previous paragraph. Now replace the S-binomial by the corresponding G chain. The reduction chain does not have any monomial higher than both the monomials of the S-binomial so no new peaks can form which is above both the monomials of S-binomial. Substitution also turns the entire chain into a G-chain and it is equivalent to the original chain. But it has one less peak or plateau at the level of the selected peak. Iterate over this step till there is no peak left. Since term-ordering is well-ordering, these iterations will have to terminate. ■

A functional definition of Gröbner basis for any ideal in the ring $k[x_1, \ldots, x_n]$ is that it is a basis of the ideal which reduces every member of the ideal to zero. We will define *pseudo Gröbner basis* in a similar fashion. In the previous section we described the reduction of a U-binomial by a set of non-balanced U-binomials. Hence the reduction of a U-binomial by set G_1 is well defined.

Definition 3.9. A U-binomial basis G of the ideal $I = \langle\, G \,\rangle$ will be called a **pseudo Gröbner basis** with respect to a given term-order if every binomial of I reduces to $0(\mathsf{mod} \langle\, G_2 \,\rangle)$ by G_1.

Algorithm 3.4 is modified Buchberger's algorithm which computes a pseudo Gröbner basis for any ideal having a U-binomial basis B. The first loop of the algorithm terminates since the initial ideal of $\langle G_1 \rangle$ strictly increases in each iteration and the underlying ring is Noetherian.

Let us now focus on the second part of the algorithm. In line 23, r is added to H if no monomial in H divides r. So, each addition to H strictly increases the ideal generated by H. Once again ring being Noetherian, this expansion of H must stop. Hence the algorithm terminates.

Theorem 3.10. *Algorithm 3.4 computes a pseudo Gröbner basis of $\langle B \rangle$ with respect to the given term ordering.*

Proof. Let (G_1, G_2) be the output of algorithm 3.4. Let $G = G_1 \bigcup G_2$. The S-polynomials of a pair of binomials in the ideal also belong to the ideal. Similarly the G_1 reduction of a binomial of the ideal also belongs to the ideal. Hence the ideal remains fixed during the computation, i.e., $\langle B \rangle = \langle G \rangle$.

In order to show that (G_1, G_2) is a pseudo-Gröbner basis of $\langle G \rangle$ we need to show that G_1 reduces every polynomial of $\langle G \rangle$ to a polynomial in $\langle G_2 \rangle$. Due to Theorem 3.3 it is sufficient to show that G_1 reduces every G-chain binomial to a polynomial in $\langle G_2 \rangle$.

Let $\mathsf{S}_{\prec}(b_1, b_2)$ be the S-polynomial of some $b_1, b_2 \in G$. Then it is itself a $G \bigcup \{\mathsf{S}_{\prec}(b_1, b_2)\}$-chain (i.e., a chain of only one binomial). The reduction chain of $\mathsf{S}_{\prec}(b_1, b_2)$ is a G-chain since $\overline{\mathsf{S}_{\prec}(b_1, b_2)}^{G_1}$ belongs to G. From Lemma 3.8 every G-chain has an equivalent bitonic G-chain.

Consider an arbitrary G-chain binomial $b = (u_1/u'_1)\mathbf{x}^{\alpha_1} + (u_2/u'_2)\mathbf{x}^{\alpha_2}$. From the previous paragraph we know that there is a bitonic G-chain with b as its chain binomial. Let C_1, C_2 and C_3 be its descending, horizontal, and ascending sections. So the C_1 and C_3^r (reverse of C_3) are reduction chains of $(u_1/u'_1)\mathbf{x}^{\alpha_1}$ and $(u_2/u'_2)\mathbf{x}^{\alpha_2}$ respectively. Let their reduced terms be $(v_1/v'_1)\mathbf{x}^{\beta_1}$ and $(v_2/v'_2)\mathbf{x}^{\beta_2}$ respectively. Then the chain-binomial of C_2 is $b' = (-v_1/v'_1)\mathbf{x}^{\beta_1} + (-v_2/v'_2)\mathbf{x}^{\beta_2}$. Since all balanced binomials of G belong to G_2, C_2 is a G_2-chain and $b' \in \langle G_2 \rangle$. ■

3.6 Saturation with respect to x_i

In this section, we we will prove a result similar to Lemma 12.1 of (Stu95) which will result into an algorithm to compute $\langle B \rangle : x_i^\infty$ efficiently.

Algorithm 3.4: $\mathcal{A}_1$: Modified Buchberger's algorithm

```
Data: $B = \{b_1, \ldots, b_s\} \subseteq S[x_1, \ldots, x_n]$ be a set of $U$-binomials; A term
      order $\prec$
Result: A pseudo Gröbner basis $(G_1, G_2)$ for $\langle\ B\ \rangle$ with respect to $\prec$.
$G_2 \leftarrow$ balanced members of $B$; $G_1 \leftarrow B \setminus G_2$ ;
repeat
    $G_{1,\text{old}} \leftarrow G_1$ ;
    for each pair $b_1, b_2 \in G_{1,old}$ such that $b_1 \neq b_2$ do
        $r \leftarrow \overline{\mathsf{S}_{\prec}(b_1, b_2)}^{G_1}$ ;
        if $r$ is non-balanced then
            $G_1 \leftarrow G_1 \bigcup \{r\}$ ;
        else
            if $r \neq 0$ then
                $G_2 \leftarrow G_2 \bigcup \{r\}$
            end
        end
    end
until $G_{1,old} = G_1$;
$H_2 \leftarrow \emptyset$;
for each $b$ in $G_2$ do
    $H \leftarrow \{b\}$;
    repeat
        $H_{\text{old}} \leftarrow H$;
        for each $b \in H_{old}$ and $b_1 \in G_1$ do
            $r \leftarrow \overline{\mathsf{S}_{\prec}(b, b_1)}^{G_1}$;
            if no any monomial of $H$ divides $r$ then
                $H \leftarrow H \bigcup \{r\}$ ;
            end
        end
    until $H_{old} = H$;
    $H_2 \leftarrow H_2 \bigcup H$;
end
return $(G_1, H_2)$ ;
```

Theorem 3.11. *Let (G_1, G_2) be the pseudo Gröbner basis of a homogeneous U-binomial ideal I in $S[x_1, \ldots, x_n]$ with respect to any graded reverse lexicographic term order in which x_i is least. Then $(G'_1 = G_1 \div x_i^\infty, G'_2 = G_2 \div x_i^\infty)$ is a pseudo Gröbner basis of $I : x_i^\infty$.*

Proof. From Theorem 3.3 we know that every polynomial f in I can be expressed as a sum of G-chain binomials and their monomials are monomials of f. So it is sufficient to show that for each G-chain binomial b, $b' = b \div x_i^\infty$ is a G'-chain binomial.

Let

$$b = \frac{u_1}{u'_1}\mathbf{x}^{\alpha_1} + \frac{u_2}{u'_2}\mathbf{x}^{\alpha_2}$$

be a G-chain binomial. From Lemma 3.8, there is a bitonic G-chain for b, say,

$$\left(\frac{v_1}{v'_1}\mathbf{x}^{\beta_1}b_1, \ \ldots \ , \ \frac{v_k}{v'_k}\mathbf{x}^{\beta_k}b_k\right).$$

Hence every monomial in the chain is less than either $\mathbf{x}^{\alpha_1}$ or $\mathbf{x}^{\alpha_2}$. Let a be the largest integer such that x_i^a divides b, i.e., x_i^a divides $\mathbf{x}^{\alpha_1}$ and $\mathbf{x}^{\alpha_2}$. Since the term ordering is graded reverse lexicographic with x_i least, x_i^a must divide every monomial of the chain. Hence there exists β'_j such that $(\mathbf{x}^{\beta_j}b_j) \div x_i^a = \mathbf{x}^{\beta'_j}(b_j \div x_i^\infty)$. So

$$b \div x_i^\infty = b \div x_i^a = \sum_j \frac{v_j}{v'_j}\mathbf{x}^{\beta'_j}\left(b_j \div x_i^\infty\right),$$

$$\text{and } \left(\frac{v_1}{v'_1}\mathbf{x}^{\beta'_1}\left(b_1 \div x_i^\infty\right), \ \ldots \ , \ \frac{v_k}{v'_k}\mathbf{x}^{\beta'_k}\left(b_k \div x_i^\infty\right)\right)$$

is a chain with chain-binomial equal to $b \div x_i^\infty$. Thus $b \div x_i^\infty$ is a G'-chain binomial. ■

3.7 Final Algorithm

Let R_0 be a commutative Noetherian ring with unity, and $U_0 \subset R_0$ be a multiplicatively closed set with unity but without zero. Let the set U_0^+ be defined as

$$U_0^+ = \{\ u \mid u \in U_0, \text{ or } -u \in U_0, \text{ or } u = 0\ \}.$$

Let S_0 denote the localization of R_0 w.r.t U_0, i.e., $S_0 = R_0[U_0^{-1}]$. Here we define a few notations to simplify the description of the algorithm. Let U_i be the set of all monomials in $x_1, \dots, x_i$ and $S_i = S_0[x_1, \dots, x_i][U_i^{-1}]$.

Let $f(\mathbf{x})$ be a polynomial in $S_i[x_{i+1}, \dots, x_n]$. Let r be the largest integer such that x_i^r occurs in the denominators of one or more terms of f. Then $x_i^\infty * f(\mathbf{x})$ denotes $x_i^r * f(\mathbf{x})$. If B is a set of polynomials of $S_i[x_{i+1}, \dots, x_n]$, then $x_i^\infty * B$ denotes $\{x_i^\infty * f(\mathbf{x}) \ : \ f(\mathbf{x}) \in B\}$.

We will be dealing with several polynomial rings simultaneously. In case of ambiguity about the underlying ring, we will denote the ideal generated by a set of polynomials B in a ring $S[x_1, \dots, x_n]$ by $\langle B \rangle_{S[x_1,\dots,x_n]}$.

Our algorithm is based on the following identities. Here B is a finite set of polynomials in $S_0[x_1, \dots, x_n]$ and B_i be the generating set of
$\langle\ B\ \rangle_{S_n} \bigcap S_i[x_{i+1}, \dots, x_n]$, $\forall i$.

Lemma 3.12. *(i)* $\langle B \rangle_{S_0[x_1,\dots,x_n]} : (x_1 \cdots x_n)^\infty = \langle B \rangle_{S_n} \bigcap S_0[x_1, \dots, x_n]$.

(ii) $\langle B \rangle_{S_n} \bigcap S_{i-1}[x_i, \dots, x_n] = \langle x_i^\infty * B_i \rangle_{S_{i-1}[x_i,\dots,x_n]} : (x_i)^\infty$

Proof. (i) Let $f \in \langle\ B\ \rangle_{S_n} \bigcap S_0[x_1, \dots, x_n]$. Hence,

$$f = \sum_j \frac{r_j}{u'_j} \frac{\mathbf{x}^{\alpha_j}}{\mathbf{x}^{\beta_j}} b_j,$$

where $b_j \in B$. So

$$\mathbf{x}^{(\beta_1+\beta_2+\dots)} \cdot f = \sum_j \left(\frac{r_j}{u'_j} \mathbf{x}^{(\alpha_j+\beta_1+\dots+\beta_{j-1}+\beta_{j+1}+\dots)} \cdot b_j \right) \in \langle\ B\ \rangle_{S_0[x_1,\dots,x_n]}.$$

Since $f \in S_0[x_1, \dots, x_n]$, $f \in \langle B \rangle_{S_0[x_1,\dots,x_n]} : (x_1 \dots x_n)^\infty$.

Conversely, let $f \in \langle\ B\ \rangle_{S_0[x_1,\dots,x_n]} : (x_1 \cdots x_n)^\infty$. Then,

$$\exists \mathbf{x}^\beta, \ \mathbf{x}^\beta f = \sum_i \frac{r_i}{u'_i} \mathbf{x}^{\alpha_i} b_i,$$

where $b_i \in B$. So $f = \sum_i (\mathbf{x}^{\alpha_i}/\mathbf{x}^\beta) b_i \in \langle B \rangle_{S_n}$. Hence, $f \in \langle B \rangle_{S_n} \bigcap S_0[x_1, \dots, x_n]$.

(ii) From the definition of $S_{j-1}[x_j, \dots, x_n]$, we have

$$\langle\ B\ \rangle_{S_n} \bigcap S_{i-1}[x_i, \dots, x_n] \subseteq \langle\ B\ \rangle_{S_n} \bigcap S_i[x_{i+1}, \dots, x_n].$$

Now, let $f \in \langle\ B\ \rangle_{S_n} \bigcap S_{i-1}[x_i, \ldots, x_n]$, then

$$f \in \langle\ B\ \rangle_{S_n} \bigcap S_i[x_{i+1}, \ldots, x_n] = \langle\ B_i\ \rangle_{S_i[x_{i+1}, \ldots, x_n]}.$$

So,

$$f = \sum_j \left(\frac{r_j}{u'_j} \frac{\mathbf{x}^{\alpha_j}}{\mathbf{x}^{\beta_j}} \right) b_j,$$

where $b_j \in B_i$ and $\mathbf{x}^{\beta_j}$ are monomials in $x_1, \ldots, x_i$. Let m be the largest exponent of x_i in the denominators in the sum-expression. So there are integers t_i's such that

$$x_i^m f = \sum_i \frac{x_i^{t_i} \mathbf{x}^{\alpha_j}}{\mathbf{x}^{\beta_j}} \left(x_i^\infty * b_j \right).$$

This sum belongs to $\langle x_i^\infty * B_i \rangle_{S_{i-1}[x_i, \ldots, x_n]}$. So $f \in \langle x_i^\infty * B_i \rangle_{S_{i-1}[x_i, \ldots, x_n]} : (x_i)^\infty$.

Now the converse. We have

$$\begin{aligned} (x_i^\infty * B_i) &\subseteq \langle\ B\ \rangle_{S_n} \bigcap S_{i-1}[x_i, \ldots, x_n] \\ \implies \langle\ x_i^\infty * B_i\ \rangle_{S_{i-1}[x_i, \ldots, x_n]} &\subseteq \langle\ B\ \rangle_{S_n} \bigcap S_{i-1}[x_i, \ldots, x_n]. \end{aligned}$$

Now we will show that the ideal on the right hand side is saturated with respect to x_i. Let $x_i^k f \in \langle B \rangle_{S_n} \bigcap S_{i-1}[x_i, \ldots, x_n]$ where $x_i, \ldots, x_n$ are not in the denominators in f. So $(1/x_i^k)(x_i^k f) \in \langle B \rangle_{S_n}$ or $f \in \langle B \rangle_{S_n}$. Since f does not have $x_i, \ldots, x_n$ in the denominators, $f \in \langle B \rangle_{S_n} \bigcap S_{i-1}[x_i, \ldots, x_n]$.

■

Using Theorem 3.11, we compute the saturation $\langle x_i^\infty * B_i \rangle_{S_{i-1}[x_i, \ldots, x_n]} : (x_i)^\infty$. Hence the final algorithm is as follows.

The graded reverse lexicographic term order requires a homogeneous ideal, hence we require homogenization for $n \geq i > 1$ cases. In case of $i = 1$, the ideal is given to be homogeneous.

Theorems 3.10, 3.11 and Lemma 3.12 establish the correctness of this algorithm.

Theorem 3.13. *Let R_0 be Noetherian commutative ring with unity. Let $U_0 \subset R_0$ be a multiplicatively closed set. Let B be a finite set of homogeneous U_0-binomials in $S_0[x_1, \ldots, x_n]$. Then algorithm $\mathcal{A}_2$ computes a pseudo-Gröbner basis of $\langle B \rangle$: $(x_1 \cdots x_n)^\infty$.*

Algorithm 3.5: $\mathcal{A}_2$:Computation of $\langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \cdots x_n)^\infty$

```
Data: Finite set $B$ of homogeneous $U_0$-binomials in $S_0[x_1, \ldots, x_n]$.
Result: A pseudo-Gröbner basis of $\langle\ B\ \rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \cdots x_n)^\infty$
$G_1 \leftarrow \{\ b \in B \mid b$ is non-balanced $\ \}$ ;
$G_2 \leftarrow \{\ b \in B \mid b$ is balanced $\ \}$ ;
for $i \leftarrow n$ to 1 do
    if $i > 1$ then
        Homogenize $G_1$ using a new variable $z$ ;
    end
    $(G_1', G_2') \leftarrow (x_i^\infty * G_1, x_i^\infty * G_2)$ ;
    /* pseudo Gröbner Basis                                */
    $(G_1, G_2) \leftarrow \mathcal{A}_1\,(G_1, G_2,$ rev. lex order with $i$ least );
    $(G_1, G_2) \leftarrow (G_1 \div x_i^\infty, G_2 \div x_i^\infty)$ ;
    $(G_1, G_2) \leftarrow (G_1|_{z=1}, G_2|_{z=1})$ ;
end
return $(G_1, G_2)$.
```

3.8 An Application: Computing kernels

In this section, we will present an algorithm to compute the kernel of a class of polynomial homomorphisms.

Consider the polynomial ring homomorphism

$$\phi' : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m],\ x_i \mapsto y_1^{a_{i1}} \cdots y_m^{a_{im}},$$

where k is a field. The kernel of ϕ' is called a *toric ideal* and, as we have seen in Chapter 2, the computation of a toric ideal is a well studied problem.

In this section, we study the problem of computing a more general kernel. Let R_0, U_0 and U_0^+ be as before, and $S_0 = R_0[U_0^{-1}]$. Consider the ring homomorphism

$$\phi : S_0[x_1, \ldots, x_n] \to S_0[y_1, \ldots, y_m],\ x_i \mapsto \frac{u_i}{u_i'}\mathbf{y}_i^\alpha,$$

where $u_i \in U_0^+, u_i' \in U_0$. The problem is to devise an efficient algorithm to compute $\ker\phi$, the kernel of ϕ.

By permuting indices we can assume that there exists r such that $u_i/u'_i \neq 0$ for $i \leq r$ and $u_i/u'_i = 0$ for $i \geq r+1$. Define

$$\phi_r : S_0[x_1, \ldots, x_r] \to S_0[y_1, \ldots, y_m],\ x_i \mapsto \frac{u_i}{u'_i}\mathbf{y}^{\alpha_i}.$$

Then, it is easy to see that

$$\ker\phi = (\ker\phi_r) + \langle x_{r+1}, \ldots, x_n\rangle. \tag{3.1}$$

So the non-trivial part of the problem is to compute $\ker\phi_r$. Hence from now onwards, we will assume that $u_i/u'_i \neq 0$ for all $1 \leq i \leq n$. Note that, since U_0 is a multiplicatively closed set without zero, ϕ-image of any monomial is non-zero.

Now, consider the following derived homomorphism

$$\phi' : k[x_1, \ldots, x_n] \to k[y_1, \ldots, y_m],\ x_i \mapsto \mathbf{y}^{\alpha_i}.$$

Since $\ker\phi$ does not contain any monomial, it can be shown that there is a one to one correspondence between the binomials of $\ker\phi$ and those of $\ker\phi'$. Hence one way to compute $\ker\phi$ is to compute a basis of the toric ideal $\ker\phi'$ and then compute a basis of $\ker\phi$ from it. Here we will describe an alternative method.

To compute the toric ideal, a set of pure difference-binomials B_{tor} is computed from ϕ' such that $\ker\phi' = \langle B_{\text{tor}}\rangle : (x_1 \cdots x_n)^\infty$. B_{tor} is computed as follows.

Let A be a matrix in which the columns are α_i's, i.e.

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \cdots & \alpha_{m1} \\ \alpha_{12} & \alpha_{22} & \cdots & \alpha_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{1n} & \alpha_{2n} & \cdots & \alpha_{mn} \end{bmatrix}.$$

Let $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ be a complete set of integer solutions of $A\mathbf{x} = 0$, i.e., all integer solutions of these equations can be expressed as linear combination of $\mathbf{u}_j$'s with integral coefficients. The $\mathbf{u}_j$'s can be computed in polynomial time by computing *Hermite normal form* of A. Let $\mathbf{u}_j^+$ and $\mathbf{u}_j^-$ be defined as follows:

$$\mathbf{u}_j^+[k] \triangleq \begin{cases} \mathbf{u}_j[k], & \text{if } \mathbf{u}_j[k] \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

We define $\mathbf{u}_j^- \triangleq \mathbf{u}_j^+ - \mathbf{u}$. Let

$$B_{\text{tor}} = \left\{ \mathbf{x}^{\mathbf{u}_j^+} - \mathbf{x}^{\mathbf{u}_j^-} \mid 1 \leq j \leq k \right\}.$$

Then it is shown in (Stu95, Chapter 4) that kernel of ϕ_{tor} is $\langle B_{\text{tor}} \rangle : (x_1 \cdots x_n)^\infty$. Note that the only non-trivial step in the computation of a toric ideal is the saturation of $\langle B_{\text{tor}} \rangle$.

Continuing with the computation of $\ker \phi$, consider the set

$$B = \left\{ \frac{v_1}{v_1'} \mathbf{x}^{\alpha_1} + \frac{-v_2}{v_2'} \mathbf{x}^{\alpha_2} \mid \mathbf{x}^{\alpha_1} - \mathbf{x}^{\alpha_2} \in B_{\text{tor}}, \right.$$
$$\left. \phi(\mathbf{x}^{\alpha_1}) = \frac{v_2}{v_2'} \phi'(\mathbf{x}^{\alpha_1}), \phi(\mathbf{x}^{\alpha_2}) = \frac{v_1}{v_1'} \phi'(\mathbf{x}^{\alpha_2}) \right\}.$$

Observe that B is a set of U-binomials and that it has no mono-binomials. Now we will show that B is the desired initial basis, i.e., $\ker \phi = \langle B \rangle : (x_1 \cdots x_n)^\infty$.

From the construction of B it is clear that every binomial of B is in the kernel of ϕ. Further, the ϕ-image of every monomial is non-zero, so $\langle B \rangle : (x_1 \cdots x_n)^\infty \subseteq \ker \phi$.

To show the converse, we will first establish that $\ker \phi$ is a binomial ideal (i.e., it can be generated by a set of binomials), thus it would be sufficient to show that every binomial in $\ker \phi$ is in $\langle B \rangle : (x_1 \cdots x_n)^\infty$.

Assume that $\ker \phi$ is not a binomial ideal. Then there must exist a polynomial p,

$$p = c_1 \mathbf{x}^{\alpha_1} + \cdots + c_j \mathbf{x}^{\alpha_j},\ j > 2$$

which cannot be expressed as a linear combination of kernel-polynomials with fewer than k terms. Let $\phi(\mathbf{x}^{\alpha_1}) = (d_1/d_1')\mathbf{y}^\beta$ which is non-zero. So there must be another term in the polynomial whose ϕ-image has monomial $\mathbf{y}^\beta$. Without loss of generality assume that $\phi(\mathbf{x}^{\alpha_2}) = (d_2/d_2')\mathbf{y}^\beta$. Then p is the sum

$$p = \left(c_1 \mathbf{x}^{\alpha_1} - c_1 \frac{d_1}{d_1'} \frac{d_2'}{d_2} \mathbf{x}^{\alpha_2} \right) + \left(\left(c_2 + c_1 \frac{d_1}{d_1'} \frac{d_2'}{d_2} \right) \mathbf{x}^{\alpha_2} + c_3 \mathbf{x}^{\alpha_3} + \cdots + c_k \mathbf{x}^{\alpha_k} \right).$$

Both polynomials are in the kernel and their sizes are smaller than j so the assumption must be incorrect.

Consider an arbitrary binomial $b = c_1 \mathbf{x}^{\alpha_1} + c_2 \mathbf{x}^{\alpha_2} \in \ker \phi$. We need to show that it is in $\langle B \rangle : (x_1 \cdots x_n)^\infty$ to complete the proof.

As $b \in \ker\phi$, the monomials of $\phi(\mathbf{x}^{\alpha_1})$ and $\phi(\mathbf{x}^{\alpha_2})$ must be same. So $\mathbf{x}^{\alpha_1} - \mathbf{x}^{\alpha_2} \in \ker\phi'$. Since $\ker\phi'$ is known to be equal to $\langle B_{\text{tor}}\rangle : (x_1 \cdots x_n)^\infty$, there exists $\mathbf{x}^\alpha$ such that $\mathbf{x}^\alpha(\mathbf{x}^{\alpha_1} - \mathbf{x}^{\alpha_2}) \in \langle B_{\text{tor}}\rangle$. Every member of B_{tor} is a pure difference-binomial, so every binomial in $\langle B_{\text{tor}}\rangle$ is a B_{tor}-chain binomial. Let

$$\mathbf{x}^\alpha\left(\mathbf{x}^{\alpha_1} - \mathbf{x}^{\alpha_2}\right) = \mathtt{B}\left(\mathbf{x}^{\beta_1}\left(\mathbf{x}^{\alpha_{1,1}} - \mathbf{x}^{\alpha_{1,2}}\right),\ \ldots\ ,\ \mathbf{x}^{\beta_t}\left(\mathbf{x}^{\alpha_{t,1}} - \mathbf{x}^{\alpha_{t,2}}\right)\right),$$

where $(\mathbf{x}^{\alpha_{i,1}} - \mathbf{x}^{\alpha_{i,2}}) \in B_{\text{tor}}$. So $\mathbf{x}^{\beta_1}\mathbf{x}^{\alpha_{1,1}} = \mathbf{x}^\alpha\mathbf{x}^{\alpha_1}$ and $\mathbf{x}^{\beta_t}\mathbf{x}^{\alpha_{t,2}} = \mathbf{x}^\alpha\mathbf{x}^{\alpha_2}$. From the construction of B we know that for each i there exist v_{i1}/v'_{i1} and v_{i2}/v'_{i2} such that $(v_{i1}/v'_{i1})\mathbf{x}^{\alpha_{i,1}} + (-v_{i2}/v'_{i2})\mathbf{x}^{\alpha_{i,2}} \in B$. Let us denote the U_0-binomial $(v_{i1}/v'_{i1})\mathbf{x}^{\alpha_{i,1}} + (-v_{i2}/v'_{i2})\mathbf{x}^{\alpha_{i,2}}$ by b_i. So the chain binomial

$$\mathtt{B}\left(\frac{c_1 v'_{11}}{v_{11}}\mathbf{x}^{\beta_1}b_1,\ \ldots\ ,\frac{c_1 v'_{11}}{v_{11}}\mathbf{x}^{\beta_t}b_t\right) = c_1\mathbf{x}^\alpha\mathbf{x}^{\alpha_1} + c'_2\mathbf{x}^\alpha\mathbf{x}^{\alpha_2}$$

belongs to $\langle B\rangle$, for some c'_2.

We have seen that $\langle B\rangle$ is contained in $\ker\phi$ so $(c_1\mathbf{x}^\alpha\mathbf{x}^{\alpha_1} + c'_2\mathbf{x}^\alpha\mathbf{x}^{\alpha_2})$ and $(c_1\mathbf{x}^\alpha\mathbf{x}^{\alpha_1} + c_2\mathbf{x}^\alpha\mathbf{x}^{\alpha_2})$ both belong to $\ker\phi$. Hence their difference $(c_2 - c'_2)\mathbf{x}^\alpha\mathbf{x}^{\alpha_2}$ must also belong to the kernel. Since ϕ-image of a monomial is non-zero, $c_2 - c'_2 = 0$. Thus we conclude that $\mathbf{x}^\alpha \cdot b \in \langle B\rangle$ or $b \in \langle B\rangle : (x_1 \cdots x_n)^\infty$.

Lemma 3.14. *Given ϕ and B as defined in the discussion above,* $\ker\phi = \langle\ B\ \rangle : (x_1 \cdots x_n)^\infty$.

Thus the problem of computing $\ker\phi$ reduces to the computation of $\langle B\rangle : (x_1 \cdots x_n)^\infty$ which can be computed by Algorithm 3.5.

3.9 Preliminary Experimental Results

In the table given below, we present some preliminary experimental results of the application of the proposed algorithm in computing toric ideals. To apply our general algorithm to this specific case, we choose R_0 to be a field k, and U_0 to be $\{1\}$. Thus, $S_0 = k$ and the polynomial ring $S_0[x_1, \ldots, x_n]$ is simply $k[x_1, \ldots, x_n]$.

We compare our algorithm with Sturmfels' (Stu95) and the *Project and Lift* algorithm (HM09), the best algorithm known to date to compute toric ideals. As expected, the table shows that our algorithm performs much better than the Sturmfels' original algorithm, as our algorithm is specifically designed for binomial ideals.

To compare with Project and Lift algorithm, we implemented it as reported on page 19 of (HM09), without optimizations reported subsequently. 4ti2 (HM09) is the optimal implementation of their algorithm. Similar optimizations are applicable in our algorithm and it too is implemented without the same. The typical results are presented in the table given below.

Number of	Size of basis		Time taken (in sec.)		
variables	Initial	Final	Sturmfels'	Project and Lift	Proposed
8	4	186	.30	0.12	0.10
	6	597	2.61	.6	0.64
10	6	729	3.2	1.1	0.50
	8	357	2.4	.40	0.29
12	6	423	1.7	.90	0.27
	8	2695	305	60	27.2
14	10	1035	10.5	4.2	2.5

Table 3.1: Preliminary experimental results comparing Project-and-Lift and our proposed algorithm

Our intuition as to why our algorithm is doing better compared to Project and Lift is that, though Project and Lift does a large part of its calculations in rings of variables less than n, it still uses Sturmfels' saturation algorithm as a subroutine, though the extent it uses the algorithm depends on the input ideal. On the other hand, our algorithm computes all saturations by the same approach.

Chapter 4

A Divide-and-Conquer Method to Compute Binomial Ideals

4.1 Introduction

The symbolic projection introduced in Chapter 2 was refined in Chapter 3 by using localized rings. We continue to refine this approach in this chapter. We develop a *divide and conquer* technique in which an ideal is mapped to ideals in lower (less variable) rings and then the results computed from those ideals are lifted back to the original rings and combined to compute the result of the original problem. This method is applied for the computation of saturation, radical, minimal primes, and cellular decomposition of binomial ideals.

This chapter is based on the work (KM12). The chapter has been arranged as follows. Section 2 deals with some basic facts about rings and ideals, and discusses irreducible and primary decompositions in the context of Noetherian rings. In the next section, we define two maps from ideals of $k[x_1, \ldots, x_n]$ to the ideals of the derived rings in $n-1$ variables, and state some useful properties. These two maps form the basis of the reduction of the problem into the subproblems. Section 4 contains the main contribution of this chapter – discussion of the proposed divide-and-conquer framework. In Section 5, we use this framework to compute radical, cellular decomposition, minimal primes, and saturation of binomial ideals.

4.2 Rings and Ideal Basics

In this chapter, we will only consider commutative rings with unity. In this section, we review a few well known definitions and results about rings and ideals which will be used later in the chapter. Also, note that k will denote an algebraically closed field.

Definition 4.1. An ideal I of a ring R is **prime**, if I is a proper ideal, and $fg \in I$ implies $f \in I$ or $g \in I$.

Definition 4.2. **Radical** of an ideal I is an ideal given by

$$\sqrt{I} = \{\ r \mid r^m \in I,\ m \geq 0\ \}.$$

An ideal is said to radical, if it is its own radical.

We now have two simple observations regarding radical ideals.

Observation 10. *Prime ideals are radical.*

Observation 11. *$I_1 \subseteq I_2$ implies that $\sqrt{I_1} \subseteq \sqrt{I_2}$.*

Definition 4.3. A ring is said to be **Noetherian** if every strictly ascending chain of ideals in the ring

$$I_1 \subsetneq I_2 \subsetneq I_3 \subsetneq \ldots$$

terminates.

The next observation presents an alternate view of Noetherian rings.

Observation 12. *A ring is Noetherian if and only if every ideal of the ring is finitely generated.*

Definition 4.4. The quotient ring

$$k[x_1, \ldots, x_n, y_1, \ldots, y_m]/\langle\ x_1y_1 - 1, \ldots, x_my_m - 1\ \rangle,$$

for $1 \leq m \leq n$, is called a **partial Laurent polynomial ring** and it is denoted by $k[x_1, \ldots, x_n, x_1^{-1}, \ldots, x_m^{-1}]$, where x_i^{-1} corresponds to y_i for $1 \leq i \leq m$. If $m = n$, then it is called a **Laurent polynomial ring**.

We now make an observation associating localization and Laurent polynomial rings.

Observation 13. *Let $R = k[x_1, \ldots, x_n]$ and U be the set of all monomials generated by the variables $\{x_1, \ldots, x_m\}$, $1 \leq m \leq n$. Then, $R[U^{-1}]$ is isomorphic to the partial Laurent polynomial ring $k[x_1, \ldots, x_n, x_1^{-1}, \ldots, x_m^{-1}]$. It is also isomorphic to $R'[x_{m+1}, \ldots, x_n]$ where R' is the Laurent polynomial ring $k[x_1, \ldots, x_m, x_1^{-1}, \ldots, x_m^{-1}]$*

Lemma 4.5. *(Eis95, Corollary 2.3) A localization of a Noetherian ring is Noetherian.*

The above lemma and the fact that polynomial rings are Noetherian imply that partial Laurent polynomial rings are also Noetherian.

For convenience in describing the algorithm in section 4.4, we present an alternative notation for partial Laurent polynomial rings. Let V be the set of variables $\{x_1, x_2, \ldots, x_n\}$, and $L = \{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ be a subset of V. Then, we will denote the partial Laurent polynomial ring $k[x_1, \ldots, x_n, x_{i_1}^{-1}, \ldots, x_{i_m}^{-1}]$ by the tuple (k, V, L).

4.2.1 Irreducible decompositions

Definition 4.6. (CLO07) Let R be a ring. An ideal $I \subseteq R$ is said to be **irreducible** if

$$I = I_1 \bigcap I_2 \implies I = I_1 \text{ or } I = I_2.$$

Definition 4.7. An **irreducible decomposition** of an ideal I is an expression of I as the intersection of irreducible ideals.

Lemma 4.8. *If an ideal I does not have an irreducible decomposition, then $\exists$ an ideal $J \supsetneq I$ which also does not have an irreducible decomposition.*

Proof. Let I be an ideal which does not have an irreducible decomposition. This also means that I is not irreducible. Consider the set of decompositions of I as the intersection of two ideals. This is certainly non-empty as it has $I \bigcap R$. Since I is not irreducible, it has a decomposition $I = I_1 \bigcap I_2$ such that both of them properly contain I. Moreover, at least one of I_1 and I_2, call it J, does not have an irreducible decomposition, otherwise I will have an irreducible decomposition. J is the desired ideal. ■

Theorem 4.9. *Every ideal in a Noetherian ring has an irreducible decomposition.*

Proof. If not, then using Lemma 4.8, we can build an strict ascending chain of ideals, each of which is not expressible as the intersection of irreducible ideals. But this is not possible as the ring is Noetherian. ■

4.2.2 Primary Ideals

Definition 4.10. An ideal I in a ring R is said to be **primary** if $fg \in I$ implies either $f \in I$ or $g^n \in I$, for some $n > 0$. Equivalently, I is **primary** if $fg \in I$ implies that either $f^m \in I$ or $g^n \in I$ for some $m, n > 0$.

Lemma 4.11. *Let I be an ideal in a Noetherian ring R. If $fg \in I$, then there exists an $n \geq 0$ such that $\langle\ f\ \rangle \bigcap \langle\ g^n\ \rangle \subseteq I$*

Proof. As R is Noetherian, $\exists n \geq 0$ such that $I : g^n = I : g^\infty$. Let $h \in \langle\ f\ \rangle \bigcap \langle\ g^n\ \rangle$. This implies $h = r_2 g^n = r_1 f$, where $r_1, r_2 \in R$. Here, $hg = r_2 g^{n+1} = r_1 fg \in I$. This shows that $r_2 \in I : g^{n+1} = I : g^n$ and hence $h \in I$. ■

Lemma 4.12. *Every irreducible ideal in a Noetherian ring is primary.*

Proof. Let I be an irreducible ideal, and $fg \in I$, where $f \notin I$. Using Lemma 4.11, we know that

$$(I + \langle\ f\ \rangle) \bigcap (I + \langle\ g^n\ \rangle) = I.$$

Since $f \notin I$, $I + \langle\ f\ \rangle$ is strictly larger than I. Hence $I + \langle\ g^n\ \rangle = I$, which implies that $g^n \in I$. ■

Definition 4.13. A **primary decomposition** of an ideal I is an expression of I as an intersection of primary ideals

$$I = \bigcap_{i=1}^{r} Q_i,$$

where Q_is are primary ideals. It is called **minimal** or **irredundant** if the $\sqrt{Q_i}$ are all distinct and $Q_i \not\supseteq \bigcap_{j \neq i} Q_j$.

Theorem 4.14. *Every ideal in a Noetherian ring has a primary decomposition.*

Proof. This follows from Theorem 4.9 and Lemma 4.12. ■

Lemma 4.15. *Radical of intersection of ideals is intersection of radicals of the ideals.*

Proof. Let the ideals involved be $I_1, I_2, \ldots, I_n$, and we want to show that

$$\sqrt{I_1 \bigcap I_2 \bigcap \ldots \bigcap I_n} = \sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \bigcap \sqrt{I_n}.$$

Let $f \in \sqrt{\bigcap_i I_i}$. This implies that for some $m > 0, f^m \in \bigcap_i I_i \implies f^m \in I_i, \forall i \implies f \in \sqrt{I_i}, \forall i$. Thus, $f \in \bigcap_i \sqrt{I_i}$. So, we have

$$\sqrt{I_1 \bigcap I_2 \bigcap \ldots I_n} \subseteq \sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \sqrt{I_n}.$$

To show the converse, let $f \in \bigcap_i \sqrt{I_i}$. Then, it is easy to see that there exists an $m > 0$, such that $f^m \in \bigcap_i I_i$. This implies that $f \in \sqrt{\bigcap_i I_i}$. Thus, we have

$$\sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \sqrt{I_n} \subseteq \sqrt{I_1 \bigcap I_2 \bigcap \ldots I_n}.$$

■

Lemma 4.16. *An ideal is primary iff its radical is prime.*

Proof. **(if)** Let I be an ideal such that $\sqrt{I}$ is prime. Let $fg \in I$, so $fg \in \sqrt{I}$. So, either $f \in \sqrt{I}$ or $g \in \sqrt{I}$. Hence, either $f^m \in I$ or $g^n \in I$, for $m, n \geq 0$. Thus, I is primary.

(only if) Let I be a primary ideal. Let $fg \in \sqrt{I}$ and $f \notin \sqrt{I}$. So for some $n > 0$, $f^n g^n \in I$ and $f^k \notin I$ for all k. As I is primary, there is some m such that $g^{nm} \in I$. Hence $g \in \sqrt{I}$.

■

Lemma 4.17. *If I and J are primary and $\sqrt{I} = \sqrt{J}$, then $I \bigcap J$ is also primary.*

Proof. Let $fg \in I \bigcap J$, and $f^j \notin I \bigcap J$ for all $j > 0$. We need to show that $g^n \in I \bigcap J$, for some $n > 0$. We claim that $f^i \notin I, \forall i > 0$. Otherwise, $f \in \sqrt{I} = \sqrt{J}$ or $f^m \in I \bigcap J$ for some $m > 0$, which contradicts the assumption. As $f^i \notin I, \forall i$ and I is primary, we deduce that $g^{n_1} \in I$ for some $n_1 > 0$. From a similar argument $g^{n_2} \in J$ for some $n_2 > 0$. Hence the proof. ■

Theorem 4.18. *Every ideal in a Noetherian ring has a minimal primary decomposition.*

Proof. Theorem 4.14 gives us a primary decomposition for any ideal. Repeated application of Lemma 4.17 gives us a primary decomposition such that all the radicals are distinct. Lastly, we can eliminate all the redundant ideals in the intersection to get a minimal primary decomposition. ■

Theorem 4.19. *Every radical ideal in a Noetherian ring has a prime decomposition.*

Proof. Let I be a radical ideal in a Noetherian ring. From Theorem 4.14, I has a primary decomposition

$$I = Q_1 \bigcap Q_2 \bigcap \dots \bigcap Q_n.$$

Then, applying Lemma A.3, we have

$$\sqrt{I} = \sqrt{Q_1} \bigcap \sqrt{Q_2} \bigcap \dots \bigcap \sqrt{Q_n}.$$

Now, observing that $\sqrt{Q_i}$s are prime (Lemma 4.16), we have the proof. ■

4.3 Two Ring Homomorphisms

4.3.1 Modulo Map

Let r be an element of a Noetherian ring R. Then $\theta : R \to R/\langle\ r\ \rangle$ denotes the natural homomorphism

$$\theta(a) = [a] = a + \langle\ r\ \rangle,\ \forall a \in R.$$

This induces a map Θ from the ideals in R containing r and the ideals of $R/\langle r\rangle$ as follows -

$$\Theta(I) = \{\ [a] \mid a \in I\ \},$$

where $I \subseteq R$ is an ideal containing r.

Similarly, we define a map Θ^{-1} from the ideals of $R/\langle\ r\ \rangle$ to the ideals of R containing r as follows

$$\Theta^{-1}(J) = \{\ x \mid [x] \in J\ \},$$

where $J \subseteq R/\langle\ r\ \rangle$ is an ideal.

Lemma 4.20. Θ *is a bijection.*

Proof. We will first show that for any ideal $I \subseteq R$ containing r, $\Theta^{-1}(\Theta(I)) = I$. From the definitions, we observe that $I \subseteq \Theta^{-1}(\Theta(I))$. Now let $x \in \Theta^{-1}(\Theta(I))$. So $[x] \in \Theta(I)$ and hence, there exists $s \in I$ such that $x - s = tr$, for some $t \in R$. Since $r, s \in I$, $x \in I$.

To show that $\Theta(\Theta^{-1}(J)) = J$ for every ideal J in $R/\langle\ r\ \rangle$, observe from the definitions that $J \subseteq \Theta(\Theta^{-1}(J))$. Now, let $[x] \in \Theta(\Theta^{-1}(J))$. So for some $t \in R$, $x + rt \in \Theta^{-1}(J)$. Hence $[x + rt] \in J$. But, as $[x] = [x + rt]$, so $[x] \in J$. ■

It is directly verifiable from the definitions that Θ and Θ^{-1} preserve set inclusion.

Lemma 4.21. Θ *and* Θ^{-1} *map primes to primes.*

Proof. Let I be a prime ideal of R containing r. Also, let $[x][y] \in \Theta(I)$. So $[xy] \in \Theta(I)$ and hence $xy \in I$ (Lemma 4.20). Being a prime ideal, I contains either x or y. Without loss of generality, let us assume that $x \in I$. So $[x] \in \Theta(I)$. This implies that $\Theta(I)$ is prime.

Let J be any prime ideal in $R/\langle\ r\ \rangle$. Let $I = \Theta^{-1}(J)$. Also, let $xy \in I$. Then, $[xy] = [x][y] \in J$. Since J is prime, without loss of generality we can assume that $[x] \in J$. Hence, $x \in I$, establishing that I is also prime. ■

Lemma 4.22. Θ *distributes over finite intersections. Similarly,* Θ^{-1} *also distributes over finite intersections.*

Proof. Let R be a ring and $I_1, I_2, \ldots, I_n \subseteq R$ be ideals, each containing r. We would like to show that

$$\Theta\left(\bigcap_i I_i\right) = \bigcap_i \Theta\left(I_i\right).$$

Let $[f] \in \Theta\left(\bigcap_i I_i\right)$. This implies that $\exists g \in \bigcap_i I_i$ such that $[g] = [f]$. Thus, $f = g + hr$, for some $h \in R$. So, $f \in \bigcap_i I_i$ or $f \in I_i, \forall i$. Hence $[f] \in \Theta(I_i)$ or $[f] \in \bigcap_i \Theta\left(I_i\right)$.

As for the other direction, let $[f] \in \bigcap_i \Theta\left(I_i\right)$. Hence $[f] \in \Theta\left(I_i\right)$, for all i which implies that $f \in I_i$, for all i (Lemma 4.20). So, $[f] \in \Theta\left(\bigcap_i I_i\right)$.

To prove the second claim, consider the ideal

$$E = \Theta^{-1}(J_1 \bigcap J_2 \bigcap \cdots),$$

where J_j are ideals in $R/\langle\ r\ \rangle$. Let $I_j = \Theta^{-1}(J_j)$. So, we have

$$\begin{aligned} E &= \Theta^{-1}\left(\Theta(I_1)\bigcap\Theta(I_2)\bigcap\cdots\right) \\ &= \Theta^{-1}\left(\Theta\left(I_1\bigcap I_2\bigcap\cdots\right)\right) && \text{preceding discussion} \\ &= I_1\bigcap I_2\bigcap\cdots \\ &= \Theta^{-1}(J_1)\bigcap\Theta^{-1}(J_2)\bigcap\cdots. && \text{Lemma 4.20} \end{aligned}$$

■

Lemma 4.23. *In a Noetherian ring* $\Theta(\sqrt{I}) = \sqrt{\Theta(I)}$

Proof. From Theorem 4.19, we have $I \subseteq \sqrt{I} = \bigcap_i P_i$, where P_is are primes. So, we have

$$\Theta(I) \subseteq \Theta(\sqrt{I}) = \Theta(\bigcap_i P_i) = \bigcap_i \Theta(P_i)$$

Using Lemma 4.21 and the fact that intersection of prime ideals is radical, we know that $\Theta(\sqrt{I})$ is a radical ideal. So, we have $\sqrt{\Theta(I)} \subseteq \Theta(\sqrt{I})$.

Conversely, as $\sqrt{\Theta(I)}$ is radical, we have

$$\sqrt{\Theta(I)} = \bigcap_i P_i,$$

where the P_i's are some primes in the modulo ring. So, we have $\Theta^{-1}(\sqrt{\Theta(I)}) = \bigcap_i \Theta^{-1}(P_i)$. Once again, from Lemma 4.21 and the fact that intersection of primes is a radical, we conclude that $\Theta^{-1}(\sqrt{\Theta(I)})$ is radical. Since $I \subseteq \Theta^{-1}(\sqrt{\Theta(I)})$, $\sqrt{I} \subseteq \Theta^{-1}(\sqrt{\Theta(I)})$ or $\Theta(\sqrt{I}) \subseteq \sqrt{\Theta(I)}$. ■

Lemma 4.24. $\Theta^{-1}(\langle\ [f_1], \ldots, [f_n]\ \rangle) = \langle\ f_1, \ldots, f_n\ \rangle + \langle\ r\ \rangle$

Proof. Let $f \in \Theta^{-1}(\langle\ [f_1], \ldots, [f_n]\ \rangle)$. So, we have $[f] \in \langle\ [f_1], \ldots, [f_n]\ \rangle$. So, f can be expressed as $\sum_i g_i f_i + gr$, for some g_i's and r in the ring. This shows that f belongs to the R.H.S. The other direction can be shown in a similar fashion. ■

4.3.2 Localization map

Let r be a non-zero-divisor of a Noetherian ring R. Let U denote the set of all powers of r

$$U = \left\{\ r^i \mid i \geq 0\ \right\}.$$

Since r is not nilpotent, U does not contain zero. U is also multiplicatively closed. Therefore $R[U^{-1}]$ is well defined.

Let $\phi : R \to R[U^{-1}]$ be the natural homomorphism given by

$$\phi(a) = \frac{a}{1}, \ \forall a \in R.$$

We define a map, Φ, induced by ϕ, from the ideals in R saturated w.r.t. r to the ideals of $R[U^{-1}]$ as follows

$$\Phi(I) = \langle \ \{ \ a/1 \mid a \in I \ \} \ \rangle,$$

where $I \subseteq R$ is an ideal saturated w.r.t. r, i.e., $I : r^{\infty} = I$. We now present some properties of Φ.

Lemma 4.25. *For any ideal $I \subseteq R$ saturated w.r.t. r, $x/r^n \in \Phi(I)$, for some $n \geq 0$ implies $x \in I$. Conversely, $x \in I$ implies $x/r^n \in \Phi(I), \forall n \geq 0$.*

Proof. Let $x/r^n \in \Phi(I)$. Then, by the construction of $\Phi(I)$, there exists b_i's in I such that $x/r^n = \sum_i (c_i/r^{k_i})(b_i/1)$ for some c_i's in R and non-negative k_i's. As r is a non-zero-divisor, the above identity implies that $r^m x - \sum_i r^{k_i'} c_i b_i \in I$, for suitable m, k_i''s $\in \mathbb{N}$. This implies that $r^m x \in I$, and using the fact that I is saturated with respect to r, we have $x \in I$.

To prove the converse, let $x \in I$. Then we have $\phi(x) = x/1 \in \Phi(I)$ and hence, $x/r^n \in \Phi(I)$, $\forall n \in \mathbb{N}$. ■

Now, we will define a map, Φ^{-1}, from the ideals in $R[U^{-1}]$ to the ideals in R which are saturated with respect to r.

$$\Phi^{-1}(J) = \left\{ \ a \mid \frac{a}{r^k} \in J, \ k \geq 0 \ \right\}.$$

From their respective definitions, it is trivial to see that Φ and Φ^{-1} preserve set inclusion.

Observation 14. *Φ^{-1} is a map from the ideals of $R[U^{-1}]$ to the ideals of R which are saturated with respect to r.*

Proof. Suppose $r^m c \in \Phi^{-1}(J)$. So from the definition of the map $r^m c/r^k \in J$, for some $k \geq 0$. Since J is an ideal in $R[U^{-1}]$, $c/1 \in J$. Hence, from the definition of Φ^{-1}, we have $c \in \Phi^{-1}(J)$. ■

We will now establish that Φ is a bijection.

Lemma 4.26. *$\Phi(\Phi^{-1}(J)) = J$ for all ideals J in $R[U^{-1}]$.*

Proof. Let $a/r^k \in J$. Then, $a \in \Phi^{-1}(J)$ and hence $a/1 \in \Phi(\Phi^{-1}(J))$. But $\Phi(\Phi^{-1}(J))$ is an ideal in $R[U^{-1}]$, so $a/r^k \in \Phi(\Phi^{-1}(J))$.

Now suppose $a/r^k \in \Phi(\Phi^{-1}(J))$. From Lemma 4.25, we have $a \in \Phi^{-1}(J)$ or $a/r^n \in J$, for some $n \in \mathbb{N}$. So $a/r^k \in J$. ■

Lemma 4.27. *$\Phi^{-1}(\Phi(I)) = I$ for all ideals I in R which are saturated with respect to r.*

Proof. If $a \in I$, then $a/1 \in \Phi(I)$. So, $a \in \Phi^{-1}(\Phi(I))$.

Now, suppose $a \in \Phi^{-1}(\Phi(I))$. So $a/r^k \in \Phi(I)$ for some $k \in \mathbb{N}$. From Lemma 4.25, we have $a \in I$. ■

Lemma 4.28. *Φ and Φ^{-1} map primes to primes.*

Proof. Let $I \subsetneq R$ be a prime ideal which is saturated with respect to r. We want to show that $\Phi(I)$ is prime. Let $(x/r^m)(y/r^n) = (xy)/r^{m+n} \in \Phi(I)$. So $xy \in \Phi^{-1}(\Phi(I)) = I$. Since I is prime, I contains x or y. Without loss of generality, let us assume that $x \in I$. Hence, from Lemma 4.25, we have $x/r^m \in \Phi(I)$.

Now suppose J is a prime ideal in $R[U^{-1}]$. Let $xy \in \Phi^{-1}(J)$. So for some m, we have $(xy)/r^m \in J$ or $(x/r^m)(y/1) \in J$. As J is prime, without loss of generality, let us assume that $x/r^m \in J$. This implies $x \in \Phi^{-1}(J)$. ■

Lemma 4.29. *Φ and Φ^{-1} distribute over intersections.*

Proof. Let $I_1, I_2, \dots$ be ideals in R, each saturated with respect to r. Then,

$$\begin{aligned} & \frac{x}{r^n} \in \bigcap_i \Phi(I_i) \\ \Longleftrightarrow\ & \frac{x}{r^n} \in \Phi(I_i),\ \forall i \\ \Longleftrightarrow\ & x \in \Phi^{-1}\left(\Phi(I_i)\right) = I_i,\ \forall i \\ \Longleftrightarrow\ & x \in \bigcap_i I_i \\ \Longleftrightarrow\ & \frac{x}{r^n} \in \Phi\left(\bigcap_i I_i\right). \end{aligned}$$

Next consider the ideals $J_1, J_2, \ldots$ in $R[U^{-1}]$. So,

$$\begin{aligned}\Phi^{-1}\left(\bigcap_i J_i\right) &= \Phi^{-1}\left(\bigcap_i \Phi\left(\Phi^{-1}(J_i)\right)\right)\\ &= \Phi^{-1}\left(\Phi\left(\bigcap_i \Phi^{-1}(J_i)\right)\right)\\ &= \bigcap_i \Phi^{-1}(J_i),\end{aligned}$$

where the second equality is due to the result in the previous paragraph. ■

Lemma 4.30. $\Phi(I : x^\infty) = \Phi(I) : x^\infty$, *for any* $x \notin U$.

Proof. It follows directly from the definitions. ■

Lemma 4.31. *In a Noetherian ring* $\Phi(\sqrt{I}) = \sqrt{\Phi(I)}$

Proof. The proof is identically same as that of Lemma 4.23 when Θ is replaced by Φ and references are suitably replaced. ■

Lemma 4.32. $\Phi^{-1}(\langle\ f_1/r^{a_1}, \ldots, f_n/r^{a_n}\ \rangle) = \langle\ f_1, \ldots, f_n\ \rangle : r^\infty$

Proof. Let

$$\begin{aligned} & f \in \Phi^{-1}\left(\langle\ \frac{f_1}{r^{a_1}}, \ldots, \frac{f_n}{r^{a_n}}\ \rangle\right)\\ \iff & \frac{f}{r^k} \in \langle\ \frac{f_1}{r^{a_1}}, \ldots, \frac{f_n}{r^{a_n}}\ \rangle, \text{ for some } k\\ \iff & \frac{f}{r^k} = \sum_i \frac{g_i}{r^{b_i}}\frac{f_i}{r^{a_i}}, \text{ for some } k\\ \iff & r^m f = \sum_i g_i f_i r^{m_i} \text{ for some } m, m_i\text{'s}\\ \iff & f \in \langle\ f_1, \ldots, f_n\ \rangle : r^\infty.\end{aligned}$$

■

4.4 The Algorithm

In this section, we focus on the main objective of this chapter. We present a general algorithm (Algorithm 4.1) based on *divide-and-conquer* technique which is useful

Algorithm 4.1: A framework for computing binomials ideals - A

```
Data: A ring $(k, X, L)$, where $k$ is algebraically closed, and $\mathsf{char}(k) = 0$;
      forbidden set $V \subseteq X \setminus L$; a binomial generating set $S$ of an ideal in
      the ring.
Result: $\mathsf{A}(\langle S \rangle)$
if $X = \phi$ then                                              // The ring is a field
    Nothing to do ;
else if $X = L$ then                                            // Laurent polynomial ring
    Compute $\mathsf{A}(\langle S \rangle)$ and return ;
else if $V = X \setminus L$ then                                // No more reductions
    Compute $\mathsf{A}(\langle S \rangle)$ and return ;
end
Let $x \in (X \setminus L) \setminus V$ ;
/* computing $\mathsf{A}(\Theta(\langle S \rangle + \langle x \rangle))$ and lift           */
Call A with ideal $\Theta(\langle S \rangle + \langle x \rangle)$, ring $(k, X \setminus \{x\}, L)$ and forbidden set
$V$ ;
Compute $\Theta^{-1}(\mathsf{A}(\Theta(\langle S \rangle + \langle x \rangle)))$ ;
/* computing $\mathsf{A}(\Phi(\langle S \rangle : x^{\infty}))$ and lift                   */
Call A with ideal $\Phi(\langle S \rangle : x^{\infty})$, ring $(k, X, L \bigcup \{x\})$ and forbidden set $V$ ;
Compute $\Phi^{-1}(\mathsf{A}(\Phi(\langle S \rangle : x^{\infty})))$ ;
/* computing $\mathsf{A}(f(\langle S \rangle : x^{\infty}))$                                */
Call A with ideal $f(\langle S \rangle)$, ring $(k, X, L)$ and forbidden set $V \bigcup \{x\}$ ;
/* Computing $\mathsf{A}(\langle S \rangle)$                                                */
Combine $\Theta^{-1}(\mathsf{A}(\Theta(\langle S \rangle + \langle x \rangle)))$, $\Phi^{-1}(\mathsf{A}(\Phi(\langle S \rangle : x^{\infty})))$ and
$\mathsf{A}(f(\langle S \rangle))$ to get $\mathsf{A}(\langle S \rangle)$ ;
/* Return                                                                                   */
return $\mathsf{A}(\langle S \rangle)$ ;
```

in computing several binomial ideals associated with a given binomial ideal. The algorithm takes as input the following 3 objects (i) A ring (k, X, L), (ii) A set of binomials, S, generating an ideal I, and (iii) A set of variables $V \subseteq X \setminus L$ called *forbidden* set. The objective of the algorithm is to compute $\mathsf{A}(\langle\ S\ \rangle)$, where A is some object associated with the binomial ideal I. In this chapter, we demonstrate how to use Algorithm 4.1 to solve the following 4 problems (i) Radical of I, (ii) Cellular decomposition of I, (iii) Minimal Primes of I, and (iv) Saturation of I w.r.t. all the variables in the ring.

We will restate, from the introduction, the two crucial observations behind this algorithm – (i) most computations involving binomial ideals compute Gröbner basis of certain ideals, and (ii) Buchberger's algorithm to compute Gröbner basis is very sensitive to the number of variables in the underlying polynomial ring. The motivation behind the algorithm is to divide the problem suitably into smaller subproblems, solve these subproblems in rings with less variables than the original ring, and combine these results to solve the original problem.

Let $x \in (X \setminus L) \setminus V$, and consider the maps (i) $\Theta : (k, X, L) \to (k, X \setminus \{x\}, L)$ induced by $\theta(y) = y + \langle\ x\ \rangle$, (ii) $\Phi : (k, X, L) \to (k, X, L \bigcup \{x\})$ induced by $\phi(y) = y/1$, and $U = \{\ x^i \mid i \geq 0\ \}$, and (iii) $f : (k, X, L) \to (k, X, L)$ which depends on the problem $\mathsf{A}()$. The reduction step involves the solutions of the subproblems (i) $\mathsf{A}(\Theta(I + \langle\ x\ \rangle))$, in ring $(k, X \setminus \{x\}, L)$ and forbidden set V, (ii) $\mathsf{A}(\Theta(I : x^{\infty}))$, in ring $(k, X, L \bigcup \{x\})$ and forbidden set V, and (iii) $\mathsf{A}(f(I))$ in ring (k, X, L) and forbidden set $V \cup \{x\}$. The first subproblem is in a ring with one less variable compared to the original ring. In the case of the second subproblem, Gröbner bases are not defined in the context of partial Laurent polynomial rings (k, X, L). But pseudo Gröbner bases (section 3.5), briefly restated later in this section in terms of partial Laurent polynomial rings, can effectively substitute for Gröbner bases for binomial ideal computations. The time complexity of the algorithm to compute pseudo Gröbner basis was shown to be dependent on the number of variables in $X \setminus L$. Hence, this subproblem is also justifiably "smaller".

The role of the forbidden set of variables is that reduction must not be done with respect to these variables. If $V = X \setminus L$, then the computation $A(I)$ must be either trivial or be possible through other method without the need for further reduction. In addition, the third subproblem should be such that it does not require the computation of a pseudo Gröbner basis since in this case the ring is same as in the original problem and involves no reduction in ring size. Here is a motivating example to justify the use of forbidden set. Suppose we want to compute the saturation, $I : (x_1 \cdots x_n)^\infty$, while I is already saturated w.r.t. x_1, x_2. Then reduction with these variables is futile. Hence we can put these variables in the forbidden set.

Next, the algorithm computes the inverse images of $\mathsf{A}(\Theta(I + \langle\ x\ \rangle))$ and $\mathsf{A}(\Phi(I : x^\infty))$ in the original ring (k, X, L). In the applications discussed in the next section, $\mathsf{A}(I)$ is either an ideal (as in the case of radical of I) or a set of ideals (as in the case of minimal primes of I). Hence these inverse images are well defined. Abusing the notations, we denote these inverse images respectively by $\Theta^{-1}(\mathsf{A}(\Theta(I + \langle\ x\ \rangle))$ and $\Phi^{-1}(\mathsf{A}(\Phi(I : x^\infty))$.

Finally in step 14, $\mathsf{A}(I)$ is to be constructed from these inverse images and $\mathsf{A}(f(I))$. One can easily observe that the algorithm terminates, as in each step either cardinality of X decreases, or that of L or V increases. This algorithm is a general method and can be tuned to a particular problem by specifying the following three steps in the context of that problem.

(steps 4, 6) Give the method to compute $A(I)$ in the base cases, i.e., when $V = X \setminus L$.

(step 13) Specify function f.

(step 14) Show how to combine the results of the subproblems.

In the next few subsections we show how to compute Θ, Φ, and their inverses using a generating set of the input ideal.

4.4.1 Computing Modulo

Let $L = \{y_1, \ldots, y_k\}$ and $X = \{x_1, \ldots, x_l\} \bigcup \{z\} \bigcup L$. Maps θ and Θ from $(k, X, L) \to (k, X \setminus \{z\}, L)$ are computed as follows. Consider an arbitrary polynomial in (k, X, L),

$$f = \sum_i \mathbf{x}^{\alpha_i}\mathbf{y}^{\beta_i} + \sum_j \mathbf{x}^{\alpha_j}\mathbf{y}^{\beta_j} z^{c_j}.$$

Then, $\theta(f) = \sum_i \mathbf{x}^{\alpha_i}\mathbf{y}^{\beta_i}$. Further, suppose $S \subset (k, X, L)$ is a set of binomials. Then, $\Theta(\langle\ S\ \rangle) = \langle\ \theta(f) \mid f \in S\ \rangle$. Conversely, if $S' \subset (k, X \setminus \{z\}, L)$, then $\Theta^{-1}(\langle\ S'\ \rangle) = \langle\ S' \bigcup \{z\}\ \rangle$, from Lemma 4.24.

4.4.2 Computing Localization

Consider the ring (k, X, L) as defined in the previous subsection. If $f \in (k, X, L)$, then $\phi(f) = f/1$.

Computing Φ and Φ^{-1} is also easy. For any $S \subset (k, X, L)$,

$$\Phi(\langle\ S\ \rangle) = \langle\ \{\ f/1 \mid f \in S\ \}\ \rangle \subseteq \left(k, X, L \bigcup \{x\}\right).$$

In the reverse direction, for any $S' \subset (k, X, L \bigcup \{z\})$, we define $\Phi^{-1}(\langle\ S'\ \rangle)$ as follows. Let

$$S' = \left\{\left(\frac{1}{z^{a_1}}\right) f_1, \ldots, \left(\frac{1}{z^{a_k}}\right) f_k\right\}$$

where f_i has no z-monomial in the denominator. Then

$$\Phi^{-1}(\langle\ S'\ \rangle) = \langle\ f_1, \ldots, f_k\ \rangle : z^\infty \subseteq \left(k, X, L \bigcup \{z\}\right).$$

Hence, a saturation computation is required to compute the basis of $\Phi^{-1}(S')$. The correctness follows from Lemmas 4.25 and 4.32.

To see how we can compute saturation with respect to z in a partial Laurent polynomial ring, we briefly revisit the results on *pseudo-Gröbner basis* in section 3.5. The results from that section are restated in the context of partial Laurent polynomial rings.

4.4.3 pseudo-Gröbner Basis

Gröbner bases are defined for ideals in rings $k[x_1, \ldots, x_n]$ (Appendix B). This notion has been generalized for binomial ideals in partial Laurent polynomial rings, called pseudo-Gröbner bases in section 3.5. Here we reproduce some relevant results in terms of partial Laurent rings.

Definition 4.33. A binomial $a\mathbf{x}^\alpha + b\mathbf{x}^\beta \in (k, X, L)$ is said to be **balanced** if $x_i \in X \setminus L$ implies $\alpha_i = \beta_i$.

Definition 4.34. For every finite binomial set G, G_1 and G_2 will denote its partition, where the former will represent the set of non-balanced binomials and the latter will represent the set of balanced binomials of G.

Definition 4.35. A binomial basis $G = (G_1, G_2)$ of a binomial ideal I will be called a pseudo Gröbner basis with respect to a given term-order if G_1 reduces every binomial of I to $0(\mathsf{mod}(G_2))$.

Theorem 4.36 (Theorem 3.10). *Every binomial ideal in (k, X, L) has a Gröbner basis with respect to any term-order.*

The Buchberger's algorithm to compute Gröbner basis has been adopted to compute pseudo-Gröbner basis in Algorithm 3.4. Finally, the following theorem shows that saturation can be computed in similar way as in $k[x_1, \ldots, x_n]$.

Theorem 4.37 (Theorem 3.11). *Let (G_1, G_2) be a pseudo Gröbner basis of a homogeneous binomial ideal in (k, X, L) with respect to a graded reverse lexicographic term order with the variable $x_i \notin L$ being the least. Then $(G_1' = G_1 \div x_i^\infty, G_2' = G_2 \div x_i^\infty)$ is a pseudo Gröbner basis of $I : x_i^\infty$.*

Here $S \div x^\infty$ is the result of the division of each polynomial in S by the largest possible power of x.

4.5 Computing $\mathsf{A}(I)$

As mentioned in the previous section, we will describe the steps 4, 6, 13 and 14 of the algorithm in context of four problems – (i) radical of a binomial ideal, (ii) cellular decomposition of a binomial ideal, (iii) the minimal prime ideals of a binomial ideal, and (iv) the saturation of a binomial ideal with respect to all variables in the ring.

4.5.1 Radical Ideal

Theorem 4.38. *Let R be a Noetherian ring, $r \in R$ a nonzero divisor, and $I \subseteq R$ be an ideal. Then,*

$$\sqrt{I + \langle\, r \,\rangle} \bigcap \sqrt{I : r^\infty} = \sqrt{I},$$

for some $r \in R$.

Proof. From Theorem 4.19, we know that every radical in a Noetherian ring has a prime decomposition. Let the prime decomposition of $\sqrt{I}$ be

$$\sqrt{I} = P_1 \bigcap P_2 \bigcap \ldots \bigcap P_n.$$

Let the collection of the primes in the decomposition be denoted by $\mathcal{P}$. Define two ideals

$$\mathcal{P}_r = \left(\bigcap_{r \in P \in \mathcal{P}} P \right), \overline{\mathcal{P}_r} = \left(\bigcap_{r \notin P \in \mathcal{P}} P \right)$$

It is easy to see that $I + \langle\, r\, \rangle \subseteq \mathcal{P}_r$. Hence, $\sqrt{I + \langle\, r\, \rangle} \subseteq \mathcal{P}_r$. Next, we want to show that $\sqrt{I : r^\infty} \subseteq \overline{\mathcal{P}_r}$.

Let $f \in I : r^\infty$. Then, $r^n f \in I$ for some $n \geq 0$. This implies that for all $P \in \mathcal{P}$, $r^n f \in P$. In particular, if $r \notin P$, then $f \in P$. We deduce that $I : r^\infty \subseteq \overline{\mathcal{P}_r}$, and hence $\sqrt{I : r^\infty} \subseteq \overline{\mathcal{P}_r}$. Putting the two observations together we have

$$\sqrt{I + \langle\, r\, \rangle} \bigcap \sqrt{I : r^\infty} \subseteq \mathcal{P}_r \bigcap \overline{\mathcal{P}_r} = \sqrt{I}$$

The converse containment $\sqrt{I} \subseteq \sqrt{I + \langle\, r\, \rangle} \bigcap \sqrt{I : r^\infty}$ is obvious. ■

This theorem leads to the following result which will help us in the execution of step 14.

Theorem 4.39. *Let R be an Noetherian ring, $r \in R$ a nonzero divisor, and $I \subseteq R$ be an ideal. Then,*

$$\sqrt{I} = \Theta^{-1}\left(\sqrt{\Theta\left(I + \langle\, r\, \rangle\right)}\right) \bigcap \Phi^{-1}\left(\sqrt{\Phi\left(I : r^\infty\right)}\right).$$

Proof. From Lemmas 4.20 and 4.23,

$$\sqrt{I + \langle\, r\, \rangle} = \Theta^{-1}\left(\Theta\left(\sqrt{I + \langle\, r\, \rangle}\right)\right) = \Theta^{-1}\left(\sqrt{\Theta\left(I + \langle\, r\, \rangle\right)}\right).$$

From Lemmas 4.27 and 4.31,

$$\sqrt{I : r^\infty} = \Phi^{-1}\left(\Phi\left(\sqrt{I : r^\infty}\right)\right) = \Phi^{-1}\left(\sqrt{\Phi\left(I : r^\infty\right)}\right).$$

So, from Theorem 4.38, we have

$$\sqrt{I} = \Theta^{-1}\left(\sqrt{\Theta\left(I + \langle\, r\, \rangle\right)}\right) \bigcap \Phi^{-1}\left(\sqrt{\Phi\left(I : r^\infty\right)}\right).$$

■

Note that we will not use the $f(I)$ branch of the reduction for this problem. Thus, Theorem 4.39 shows that the *combine* step (step 14) is the computaton of an intersection. Also, we will have $V = \emptyset$. The base case computation in step 4 of the algorithm is trivial because all binomial ideals in a Laurent polynomial ring are already radical as shown below.

Theorem 4.40. *(ES96, Corollary 2.2) Let J be a binomial ideal in the ring (k, X, ϕ). Then, if k is algebraically closed and $\mathsf{char}(k) = 0$, then $J : (\Pi_{x\in X} x)^\infty$ is radical.*

Corollary 4.41. *Let k be an algebraically closed field, with $\mathsf{char}(k) = 0$. Then, all binomial ideals in (k, X, X) are radical.*

Proof. Let J be a binomial ideal in the ring (k, X, X), where $X = \{x_1, \ldots, x_n\}$. Consider the ideal localization map, Φ_n, from $(k, X, X \setminus \{x_n\})$ to (k, X, X). Under this map, we know that $\Phi_n^{-1}(J)$ is saturated w.r.t x_n. Similarly, if we consider the map Φ_{n-1} from $(k, X, X\setminus\{x_{n-1}, x_n\})$ to $(k, X, X\setminus\{x_n\})$, then the ideal $\Phi_{n-1}^{-1}(\Phi_n^{-1}(J))$ is saturated w.r.t. x_{n-1}. So we have

$$\begin{aligned} \Phi_n^{-1}(J) &= \Phi_n^{-1}(J) : x_n^\infty \\ \implies \Phi_{n-1}^{-1}(\Phi_n^{-1}(J)) &= \Phi_{n-1}^{-1}(\Phi_n^{-1}(J) : x_n^\infty) \\ &= \Phi_{n-1}^{-1}(\Phi_n^{-1}(J)) : x_n^\infty \qquad (\text{ Lemma } 4.30) \end{aligned}$$

Thus, $\Phi_{n-1}^{-1}(\Phi_n^{-1}(J))$ is saturated w.r.t. $\{x_{n-1}, x_n\}$. Continuing this argument we see that $\Phi_1^{-1}(\cdots(\Phi_n^{-1}(J))\cdots)$, in the ring (k, X, ϕ), is saturated w.r.t. $\{x_1, \ldots, x_n\}$. From the previous theorem $\Phi_1^{-1}(\cdots(\Phi_n^{-1}(J)))$ is radical. Now, by repeated application of Lemma 4.31 we deduce that J is radical too. ■

4.5.2 Cellular Decomposition

In this section we will generalize the notion of **cellular ideals** to partial Laurent polynomial rings, establish that every ideal has a cellular decomposition, and use our framework to compute such a decomposition.

Let (k, X, L) be a partial Laurent polynomial ring. For a given set of variables $\mathscr{E} \subseteq (X \setminus L)$ and a vector $d = (d_i)_{i\in(X\setminus L)\setminus\mathscr{E}}$, we define the ideal $M(\mathscr{E})^{(d)}$ as

$$M(\mathscr{E})^{(d)} := \langle \; \{ \; x_i^{d_i} \mid i \in (X \setminus L) \setminus \mathscr{E} \; \} \; \rangle.$$

Now, we are ready to generalize the definition of cellular ideals.

Definition 4.42. An ideal I of (k, X, L) is said to be **cellular**, if for some $\mathscr{E} \subseteq (X \setminus L)$, we have $I = I : \left(\prod_{i \in \mathscr{E}} x_i\right)^{\infty}$, and I contains $M(\mathscr{E})^{(d)}$ for some vector d.

Next, we will state a trivial observation characterizing cellular ideals.

Observation 15. *An ideal I is cellular iff $\exists \mathscr{E} \subseteq (X \setminus L)$ and $d = (d_i)_{i \in (X \setminus L) \setminus \mathscr{E}}$, such that*

$$I = \left(I + M(\mathscr{E})^d\right) : \left(\prod_{i \in \mathscr{E}} x_i\right)^{\infty}.$$

In such a case, we will denote I by $I_{\mathscr{E}}^{(d)}$.

This observation helps us to make the following claim regarding cellular ideals and Φ^{-1}.

Lemma 4.43. Φ^{-1} *preserves cellular ideals.*

Proof. Let Φ^{-1} be a map from (k, X, L) to $(k, X, L \setminus \{x\})$, where $x \in L$, and consider the cellular ideal $I = I_{\mathscr{E}}^{(d)}$ in (k, X, L). As $\Phi^{-1}(I)$ is saturated w.r.t. x, it is a cellular ideal with $\Phi^{-1}(I) = \Phi^{-1}(I)_{\mathscr{E} \bigcup \{x\}}^{(d')}$, where d' is the same vector as d, except that it does not contain the component corresponding to x. ■

Lemma 4.44. *Let $s \in \mathbb{N}$ be such that $I : r^s = I : r^{\infty}$ in some Noetherian ring R. Then,*

$$I = (I + \langle\ r^s\ \rangle) \bigcap (I : r^s).$$

Proof. Let $h \in (I + \langle\ r^s\ \rangle) \bigcap (I : r^s)$. Then

$$\begin{aligned}
& h = i + gr^s \in I : r^s \text{ for some } i \in I, g \in R \\
\Longrightarrow\ & hr^s = ir^s + gr^{2s} \in I. \\
\Longrightarrow\ & gr^{2s} \in I \\
\Longrightarrow\ & g \in I : r^{2s} = I : r^s \\
\Longrightarrow\ & gr^s \in I \\
\Longrightarrow\ & h \in I
\end{aligned}$$

■

Now, we are ready to state how to compute a cellular decomposition of I. The computation will not use $\mathsf{A}(\Theta(I))$ branch of the reduction. $f(I)$ is defined as $I + \langle\, x^s \,\rangle$, where $s \in \mathbb{N}$ is such that $I : x^s = I : x^\infty$. We see that in this case, we add x to the forbidden set. This is done because x is a nilpotent in $I + \langle\, x^s \,\rangle$. So, whenever a variable is added to the forbidden set, it is ensured that it is a nilpotent. By using Lemmas 4.29 and 4.43, we see that cellular decomposition of $\Phi(I : x^\infty)$ gives us a cellular decomposition of $I : x^s$.

To combine the decompositions of $\mathsf{A}(I : x^s)$ and $\mathsf{A}(f(I))$, we use Lemma 4.44. In other words, cellular ideals for I are the union of those for $I : x^\infty$ and $I + \langle\, r^s \,\rangle$.

What remains is to specify the computations at the base cases, i.e., $X = L \bigcup V$. Ideals in the base cases are already cellular because the ring is localized with respect to L-variables and the variables of $V = X \setminus L$ are nilpotent of the ideals. Thus there is no computation required in steps 4 and 6.

4.5.3 Prime Decomposition

In this case, as in the computation of a radical, the $\mathsf{A}(f(I))$ branch will not be used. We will first handle the base case, i.e. how to compute the minimal primes of a binomial ideal in a Laurent polynomial ring (step 4). To do this, we will mention (without proof) a set of results from (ES96).

Definition 4.45. A **partial character** on $\mathbb{Z}^n$ is a homomorphism ρ from a sublattice L_ρ of $\mathbb{Z}^n$ to the multiplicative group k^*. A partial character will always refer to the tuple (ρ, L_ρ).

For a proper binomial ideal I in (k, X, X), let us define a partial character $(\rho, L(I))$, where

$$L(I) = \{\ \alpha \mid \mathbf{x}^\alpha - c \in I\ \}.$$

It is easy to verify that $L(I)$ is a lattice. The function ρ is given by

$$\rho(\alpha) = c, \text{ where } \mathbf{x}^\alpha - c \in I.$$

Conversely, given a partial character (ρ, L), we will define a binomial ideal as

$$I(\rho) = \langle\ \{\ \mathbf{x}^\alpha - c \mid \alpha \in L, \rho(\alpha) = c\ \}\ \rangle.$$

Theorem 4.46. *For any proper binomial ideal in (k, X, X), there is a unique partial character ρ on $\mathbb{Z}^n$ such that $I = I(\rho)$.*

Definition 4.47. If L is a sublattice of $\mathbb{Z}^n$, then the saturation of L is the lattice

$$\mathsf{Sat}(L) = \{\ m \in \mathbb{Z}^n \mid dm \in L \text{ for some } d \in \mathbb{Z}\ \}\,.$$

We can compute $\mathsf{Sat}(L)$ for any lattice L by simple change of variables in (k, X, X) (Swa11).

Definition 4.48. If (ρ, L_ρ) is a partial character, any partial character $(\rho', Sat(L_\rho))$ is called a **saturation** of (ρ, L_ρ) if ρ' coincides with ρ when restricted to L_ρ.

Theorem 4.49. *(ES96, Corollary 2.2) If g is the order of the group $\mathsf{Sat}(L_\rho)/L_\rho$, then there are g distinct saturations of ρ: $\rho_1, \ldots, \rho_g$. Also*

$$I(\rho) = \bigcap_{j=1}^{g} I(\rho_j).$$

Theorem 4.50. *(ES96, Corollary 2.6) The radical of a cellular ideal is of the form $I(\rho) + M(\mathscr{E})^{(d)}$ (d is vector with all 1s), where ρ is a partial character. Moreover, its minimal primes are the lattice ideals with the saturations of (ρ, L_ρ).*

In the base case, we have a Laurent polynomial ring. To determine the set of minimal primes of a binomial ideal $I = I(\rho)$, all we need to do is to compute the saturations of ρ. The lattice ideals corresponding to these saturations are the associated primes of $I(\rho)$. The minimal of these ideals constitute the prime decomposition.

Let us discuss how we can combine the results from the modulo and the localization branch (step 14). From the recursive calls of the algorithm we have computed the minimal primes of $\Theta(I + \langle\ r\ \rangle)$ and $\Phi(I : r^\infty)$. Let the set of minimal primes be denoted by $\mathcal{P}_\Theta$ and $\mathcal{P}_\Phi$, respectively. So, we have

$$\sqrt{\Theta(I + \langle\ r\ \rangle)} = \bigcap_{P \in \mathcal{P}_\Theta} P$$
$$\sqrt{\Phi(I : r^\infty)} = \bigcap_{P \in \mathcal{P}_\Phi} P.$$

From Theorem 4.39, we have

$$\begin{aligned}\sqrt{I} &= \Theta^{-1}\left(\sqrt{\Theta\left(I + \langle\, r\,\rangle\right)}\right) \bigcap \Phi^{-1}\left(\sqrt{\Phi\left(I + \langle\, r\,\rangle\right)}\right) \\ &= \left(\bigcap_{P\in\mathcal{P}_\Theta} \Theta^{-1}(P)\right) \bigcap \left(\bigcap_{P\in\mathcal{P}_\Phi} \Phi^{-1}(P)\right)\end{aligned}$$

We know that Θ and Φ map primes to primes (Lemmas 4.21 and 4.28). The desired set of prime ideals is $\{\ \Theta^{-1}(P) \mid P \in \mathcal{P}_\Theta\ \} \bigcup \{\ \Phi^{-1}(P) \mid P \in \mathcal{P}_\Phi\ \}$. We just need to remove the redundant ones.

4.5.4 Saturation

Suppose I is saturated with respect to $\left\{x_{i_1}, \ldots, x_{i_j}\right\}$ then we begin the computation with $V = \left\{x_{i_1}, \ldots, x_{i_j}\right\}$. For this problem, we only use the $\mathsf{A}(I : x^\infty)$ branch of the reduction. The base case for this algorithm will be $X \setminus L = V$ (step 6). As Φ preserves saturation (Lemma 4.30), the ideal is already saturated in this ring. Since the algorithm uses only one branch of the reduction, step 14 is redundant.

Index

Appendix A

Ring Basics

In this chapter, we will review a few basic definitions and results about rings and ideals. We will avoid proving the simple observations and lemmas, and give a proof sketch for the more involved results. For a more detailed discussion on the subject, the reader is advised to consult (CLO07) and (Eis95).

A.1 Rings

A **ring** is defined as an abelian group R with an operation $(a, b) \mapsto ab$ called *multiplication* and an "identity element" 1, satisfying, for all $a, b, c \in R$:

$$\begin{aligned} a(bc) &= (ab)c && \text{(associativity)} \\ \left.\begin{aligned} a(b+c) &= ab+ac \\ (b+c)a &= ba+ca \end{aligned}\right\} & && \text{(distributivity)} \\ 1a = a1 &= a && \text{(identity)} \end{aligned}$$

The ring is **commutative** if, in addition, $ab = ba$ for all $a, b \in R$. Unless otherwise stated, in the rest of the discussion, the word ring will be used to denote a commutative ring with identity.

A subset I of a commutative ring R is said to be an **ideal** in R if it satisfies

- $0 \in I$ (where 0 is the zero element of R).
- If $a, b \in I$, then $a + b \in I$.

- If $a \in I$ and $r \in R$, then $r \cdot a \in I$.

An ideal I is said to be **generated** by a subset $S \subseteq R$ if every element $t \in I$ can be written in the form

$$t = \sum_{i=1}^{n} r_i s_i$$

with r_i in R and s_i in S. We shall write $\langle\ S\ \rangle$ for the ideal generated by a subset $S \subseteq R$.

A ring is said to be **Noetherian** if every ideal of the ring is finitely generated. The following observation gives another criterion for a ring to be Noetherian.

Observation 16. *A ring is Noetherian if every strictly ascending chain of ideals of the ring terminates.*

Proof. Let R be a ring, and $I \subseteq R$ be an ideal. We want to establish that I is finitely generated. We will prove by contradiction.

Let I be such that it is not finitely generated. Consider the ascending chain of ideals

$$\{0\} \subsetneq \langle\ f_1\ \rangle \subsetneq \langle\ f_1, f_2\ \rangle \subsetneq \langle\ f_1, f_2, f_3\ \rangle \subsetneq \cdots$$

where $f_i \in I \setminus \langle\ f_1, \ldots, f_{i-1}\ \rangle$. Such an f_i can always be found, as I is not finitely generated. Thus, we have an infinite ascending chain of ideals, which is absurd as R is Noetherian. ■

A.2 Ideals

In the previous section, we have seen what ideals are. We are now going to define a few ideals. These are the kind of ideals we will be dealing with in this thesis. All the ideals are defined in the context of a commutative ring, R, with multiplicative identity

An ideal I is said to be **irreducible** if

$$I = I_1 \bigcap I_2 \text{ implies } I = I_1 \text{ or } I = I_2.$$

Later we will prove(Theorem 4.9) that any ideal can be expressed as the intersection of irreducible ideals.

An ideal I is said to be **radical** if $f^m \in I$ for any integer $m \geq 1$ implies that $f \in I$. The **radical** of I, denoted by $\sqrt{I}$, is the set

$$\sqrt{I} = \{ \ f \mid f^m \in I \text{ for some integer } m \geq 1 \ \}.$$

We state two simple properties of radicals. They follow directly from the definitions.

Lemma A.1. *If I is an ideal in R, then $\sqrt{I}$ is an ideal in R containing I. Furthermore, $\sqrt{I}$ is a radical ideal.*

Lemma A.2. *Radical preserves set inclusion.*

The next lemma deals with the relationship of radicals and intersections.

Lemma A.3. *Radical of intersection of ideals is intersection of radicals of the ideals.*

Proof. Let the ideals involved be $I_1, I_2, \ldots, I_n$, and we want to show that

$$\sqrt{I_1 \bigcap I_2 \bigcap \ldots \bigcap I_n} = \sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \bigcap \sqrt{I_n}.$$

Let $f \in \sqrt{\bigcap_i I_i}$. This implies that $f^m \in \bigcap_i I_i \implies f^m \in I_i, \forall i \implies f \in \sqrt{I_i}, \forall i$. Thus, $f \in \bigcap_i \sqrt{I_i}$. So, we have

$$\sqrt{I_1 \bigcap I_2 \bigcap \ldots I_n} \subseteq \sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \sqrt{I_n}.$$

To show the converse, let $f \in \bigcap_i \sqrt{I_i}$. Then, it is easy to see that there exists an $m \geq 0$, such that $f^m \in \bigcap_i I_i$. This implies that $f \in \sqrt{\bigcap_i I_i}$. Thus, we have

$$\sqrt{I_1} \bigcap \sqrt{I_2} \bigcap \ldots \sqrt{I_n} \subseteq \sqrt{I_1 \bigcap I_2 \bigcap \ldots I_n}.$$

■

An ideal I in a ring R is said to be **primary** if $fg \in I$ implies either $f \in I$ or $g^n \in I$, for some $n > 0$. The next lemma gives an alternate definition of primary ideals.

Lemma A.4. *I is primary if $fg \in I$ implies that either $f^m \in I$ or $g^n \in I$ for some $m, n > 0$.*

An ideal I is said to be **proper** if $I \subsetneq R$. An ideal I is **prime** if I is a proper ideal and if $f, g \in R$ and $fg \in I$ implies $f \in I$ or $g \in I$. Prime ideals will be used extensively to study radical ideals.

Observation 17. *Prime ideals are radical.*

We now define saturation of an ideal. Chapters 2 and 3 will deal with the saturation of a special kind of ideal, namely *homogeneous binomial ideal.* Let I is an ideal, and r be an element of R. We define the following sets –

$$(I : r) = \{ f \in R \mid fr \in I \}$$
$$(I : r^{\infty}) = \bigcup_{d=1}^{\infty} (I : r^d) \subseteq R.$$

The set $(I : r^{\infty})$ is called the **saturation** of I with respect to r.

Observation 18. *Both of the sets defined above are ideals.*

Observation 19. $(I : r^{\infty}) = \{ f \in R \mid fr^m \in I, m \in \mathbb{N} \}$.

A.3 Rings homomorphisms

If R and S are rings, then a ring **homomorphism** is a function from $\phi : R \to S$, such that

- $\phi(a + b) = \phi(a) + \phi(b)$ for all $a, b \in R$.
- $\phi(ab) = \phi(a)\phi(b)$ for all $a, b \in R$.
- $\phi(1) = 1$.

A homomorphism ϕ is said to be **surjective** , or a **surjection** , if every element s in S has a corresponding element r in R so that $\phi(r) = s$. A homomorphism is **injective** , or an **injection** , if for r, r' in R $\phi(r) = \phi(r') \implies r = r'$. A homomorphism is **bijective** , or a **bijection** , if ϕ is both injective and surjective. If the ring homomorphism is bijective, then it is also called an **isomorphism** . **Kernel** of a homomorphism ϕ, denoted as $\ker \phi$, is defined as -

$$\ker \phi = \{ f \in R \mid \phi(f) = 0 \} .$$

Appendix B

Gröbner basis

B.1 Introduction

As we have discussed earlier, *Gröbner bases* are central to all computational problems encountered in polynomial rings. In this chapter, we will a brief introduction to Gröbner basis and present the Buchberger's algorithm (Buc76) to compute Gröbner Basis. For a more detailed discussion on the subject, the reader is advised to consult (AL94).

B.2 Polynomial Rings

In this section, we define a special kind of a commutative Noetherian ring with identity, called the *polynomial ring*. All computations that are discussed in this thesis are done on ideals of this ring. To start with, we define some basic terminologies related to elements of the ring, and then define one of the key computational tools used in the ring, namely *the Gröbner basis*.

B.2.1 Basics

A **monomial** in $x_1, \ldots, x_n$ is a product of the form

$$x_1^{\alpha_1} \cdots x_n^{\alpha_n},$$

where all of the exponents $\alpha_1, \dots, \alpha_n$ are nonnegative integers. For simplicity, we will denote $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ by $\mathbf{x}^\alpha$.

Let k denote a field. A **polynomial** in the variables $x_1, \dots, x_n$ with coefficients in k is a finite linear combination (with coefficients in k) of monomials. We will write a polynomial f in the from

$$f = \sum_\alpha a_\alpha \mathbf{x}^\alpha, \quad a_\alpha \in k,$$

where the sum is over a finite number of n-tuples $\alpha = (\alpha_1, \dots, \alpha_n)$. The set of all polynomials in $x_1, \dots, x_n$ with coefficients in k, denoted by $k[x_1, \dots, x_n]$, is called the **polynomial ring**.

Theorem B.1 (Hilbert Basis Theorem)**.** *Every ideal $I \subseteq k[x_1, \dots, x_n]$ has a finite generating set.*

Let $f = \sum_\alpha a_\alpha \mathbf{x}^\alpha$ be a polynomial in $k[x_1, \dots, x_n]$. We have the following terminologies –

- We call a_α the **coefficient** of the monomial $\mathbf{x}^\alpha$.
- If $a_\alpha \neq 0$, then we call $a_\alpha \mathbf{x}^\alpha$ a **term** of f.

We next define an ordering on all the monomials in $k[x_1, \dots, x_n]$. We will see that an ordering forms an essential part of any algorithm in polynomial rings. In almost all cases, the termination of algorithms will be ensured by the monomial ordering.

Definition B.2. A **monomial ordering** on $k[x_1, \dots, x_n]$ is any relation $\prec$ on $\mathbb{Z}^n_{\geq 0}$, satisfying:

- $\prec$ is a total (or linear) ordering on $\mathbb{Z}^n_{\geq 0}$.
- If $\alpha \prec \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \prec \beta_\gamma$.
- $\prec$ is a **well-ordering** on $\mathbb{Z}^n_{\geq 0}$. This mean that every nonempty subset of $\mathbb{Z}^n_{\geq 0}$ has a smallest element under $\prec$.

Monomial orderings are also sometimes referred to as **term ordering** .

Lemma B.3. *An order relation $\prec$ on $\mathbb{Z}^n_{\geq 0}$ is a well-ordering if and only if every strictly decreasing sequence in $\mathbb{Z}^n_{\geq 0}$*

$$\alpha(1) \prec \alpha(2) \prec \alpha(3) \cdots$$

eventually terminates.

We provide a few examples of monomial orderings. Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$.

Lexicographic Order This monomial order is denoted by $\succ_{\text{lex}}$. We say $\alpha \succ_{\text{lex}} \beta$ if the vector difference $\alpha - \beta \in \mathbb{Z}^n_{\geq 0}$, the left-most nonzero entry is positive.

Graded Lexicographic Order Let $\vec{d}$ be a vector in $\mathbb{N}^n$. We say $\alpha \succ \beta$ if

$$\vec{d} \cdot \alpha > \vec{d} \cdot \beta, \text{ or } \vec{d} \cdot \alpha = \vec{d} \cdot \beta \text{ and } \alpha \succ_{\text{lex}} \beta.$$

Graded Reverse Lexicographic Order Let $\vec{d}$ be a vector in $\mathbb{N}^n$. We say $\alpha \succ$ if

$$\vec{d} \cdot \alpha > \vec{d} \cdot \beta, \text{ or } \vec{d} \cdot \alpha = \vec{d} \cdot \beta$$

and, in $\alpha - \beta \in \mathbb{Z}^n_{\geq 0}$, the right-most nonzero entry is negative.

In each of these orderings, one can show that they are monomial orders. We abuse the notation and say $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$ if we have $\alpha \prec \beta$. As a special mention, a graded reverse lexicographic term order with grading vector $\vec{d}$ and with x_i as the least variable will be denoted as $\prec_{\vec{d},i}$.

Let $f = \sum_\alpha a_\alpha \mathbf{x}^\alpha$ be a nonzero polynomial in $k[x_1, \ldots, x_n]$ and let $\prec$ be a monomial order.

1. The **leading monomial** of f is

$$\mathsf{inm}_\prec(f) = \mathsf{max}_\prec \{ \mathbf{x}^\alpha \mid a_\alpha \neq 0 \}$$

2. The **leading coefficient** of f is

$$\mathsf{inc}_\prec(f) = c_\beta,$$

where $\mathbf{x}^\beta = \mathsf{inm}_\prec(f)$.

3. The **leading term** of f is

$$\mathsf{in}_{\prec_{\vec{d},i+1}}(f) = \mathsf{inc}_{\prec}(f) \cdot \mathsf{inm}_{\prec}(f)$$

Sometimes, the word "initial" is used instead of "leading", hence the abbreviation "in".

Definition B.4. We define the **initial ideal** of an ideal $I \subseteq k[x_1, \ldots, x_n]$ other than $\{0\}$ as the ideal

$$\mathsf{in}_{\prec_{\vec{d},i+1}}(I) = \langle \left\{ \mathsf{in}_{\prec_{\vec{d},i+1}}(f) \mid f \in I \right\} \rangle$$

In general, if B is any subset of $k[x_1, \ldots, x_n]$, then we define

$$\mathsf{in}_{\prec_{\vec{d},i+1}}(B) = \left\{ \mathsf{in}_{\prec_{\vec{d},i+1}}(f) \mid f \in B \right\}.$$

Only in the case of an ideal, say I, $\mathsf{in}_{\prec_{\vec{d},i+1}}(I)$ will represent an ideal.

B.2.2 Polynomial Division

Let $g, g_1, \cdots, g_s$ be polynomials in $k[x_1, \ldots, x_n]$ and $\prec$ be a term order in $k[x_1, \ldots, x_n]$. Then, the polynomial expression -

$$g = \sum_i q_i g_i + r$$

is said to be a **standard expression** for g if

- $\mathsf{in}_{\prec}(q_i g_i) \preceq \mathsf{in}_{\prec_{\vec{d},i+1}}(g)$, $\forall i$
- No monomial of r is divisible by $\mathsf{in}_{\prec_{\vec{d},i+1}}(g_i)$ for any i. More formally, no monomial of r belongs to $\langle \left\{ \mathsf{in}_{\prec_{\vec{d},i+1}}(g_i) \mid 1 \leq i \leq s \right\} \rangle$.

Here, r is called the **remainder** and q_i's are called the **quotients** of the division of g by $\{g_1, \ldots, g_s\}$.

Expressing a polynomial g as a standard expression in terms of a set of polynomials $G = \{g_1, \ldots, g_s\}$ is also known as the **division** of g by G or, the **reduction** of g by G. This is denoted by $\overline{g}^G$. Theorem B.5 establishes that for every pair of polynomial and set of polynomials, standard expression exists. Algorithm B.1 gives an algorithm to compute such an expression.

Theorem B.5 (Division Algorithm). *Fix a monomial order $\prec$ on $\mathbb{Z}^n_{\geq 0}$, and let $F = (f_1, \ldots, f_s)$ be an ordered s-tuple of polynomials in $k[x_1, \ldots, x_n]$. Then every $f \in k[x_1, \ldots, x_n]$ can be written as*

$$f = a_1 f_1 + \cdots + a_s f_s + r, \tag{B.1}$$

where $a_i, r \in k[x_1, \ldots, x_n]$, and either $r = 0$ or r is a linear combination, with coefficients in k, of monomials, none of which is divisible by any of $\mathsf{in}_{\prec_{\vec{d},i+1}}(f_1), \ldots,$ $\mathsf{in}_{\prec_{\vec{d},i+1}}(f_s)$. Furthermore, if $a_i f_i \neq 0$, then we have

$$\mathsf{in}_{\prec_{\vec{d},i+1}}(f) \succeq \mathsf{in}_{\prec_{\vec{d},i+1}}(a_i f_i).$$

Algorithm B.1: $\mathsf{Division}(f, \{f_1, \ldots, f_s\}, \prec)$

Data:

- A polynomial f
- A set $B = \{f_1, \ldots, f_s\} \subseteq k[x_1, \ldots, x_n]$
- a term order $\prec$ over $k[x_1, \ldots, x_n]$

Result: $\overline{f}^B$

```
$p \leftarrow f$ ;
$r \leftarrow 0$ ;
while $p \neq 0$ do
    if $\exists i$ such that $\mathsf{in}_{\prec_{\vec{d},i+1}}(f_i) \mid \mathsf{in}_{\prec_{\vec{d},i+1}}(p)$ then
        $p \leftarrow \frac{1}{\mathsf{inc}_{\prec}(f_i)} \mathsf{S}(p, f_i, \prec)$
    else
        $r \leftarrow r + \mathsf{in}_{\prec_{\vec{d},i+1}}(p)$ ;
        $p \leftarrow p - \mathsf{in}_{\prec_{\vec{d},i+1}}(p)$ ;
    end
end
return $r$ ;
```

B.2.3 Gröbner Basis

We will now discuss one of the most important tools used to compute numerous ideals in the polynomial ring – *the Gröbner basis.* In almost all of the computations, we would find that computing Gröbner basis is the most time consuming step. As a consequence, there has been numerous efforts to speed-up Gröbner basis computation (Fau99; Fau02). In this section, we will discuss one of the most popular ways of computing Gröbner basis, namely *Buchberger algorithm.* For a more detailed discussion on the subject, the reader is directed towards (AL94).

Definition B.6. For a given monomial order $\prec$, a finite subset G of an ideal I is said to be a **Gröbner basis** if

$$\langle \, \mathsf{in}_{\prec_{\vec{d},i+1}}(G) \, \rangle = \mathsf{in}_{\prec_{\vec{d},i+1}}(I) \, .$$

Gröbner basis of an ideal I w.r.t. the monomial order $\prec$ will be denoted by $\mathcal{G}_{\prec}(I)$

Lemma B.7. *Gröbner basis for an ideal I is a basis of I.*

Before describing the algorithm to compute Gröbner basis, we will need a few more definitions. Let α and β be two vectors in $\mathbb{N}^n$, and let $\alpha[i]$ and $\beta[i]$ denote their i^{th} components, respectively. Then, by $\alpha \vee \beta$, we denote the vector whose i^{th} component is given by -

$$(\alpha \vee \beta)[i] \triangleq \mathsf{max}\left\{\alpha[i], \beta[i]\right\} .$$

This is also called the LCM of α and β. With a little abuse of notation, we will use

$$\mathbf{x}^{\alpha} \vee \mathbf{x}^{\beta} \triangleq \mathbf{x}^{(\alpha \vee \beta)}.$$

Let $\prec$ denote a term order in $k[x_1, \ldots, x_n]$. Consider any two polynomials, $h_1, h_2 \in k[x_1, \ldots, x_n]$. Let

$$c_1\mathbf{x}^{\alpha_1} = \mathsf{in}_{\prec_{\vec{d},i+1}}(h_1) \, , \text{ and } c_2\mathbf{x}^{\alpha_2} = \mathsf{in}_{\prec_{\vec{d},i+1}}(h_2)$$

be the leading terms of h_1 and h_2, respectively. We now define two vectors β_1 and β_2 as –

$$\beta_1 = (\alpha_1 \vee \alpha_2) - \alpha_1, \text{ and } \beta_2 = (\alpha_1 \vee \alpha_2) - \alpha_2.$$

Then the **S-polynomial** of h_1, h_2 is defined as –

$$\mathsf{S}(h_1, h_2) = c_2\mathbf{x}^{\beta_1}h_1 - c_1\mathbf{x}^{\beta_2}h_2.$$

Observe that, if $\mathsf{in}_{\prec_{\vec{d},i+1}}(h_2)$ divides $\mathsf{in}_{\prec_{\vec{d},i+1}}(h_1)$, then $\mathsf{S}(h_1, h_2)$ is the first step in the reduction of h_1 by h_2.

We now state a criterion, known as the *Buchberger's Criterion* (Theorem B.8), which a basis of an ideal must satisfy to be a Gröbner basis. This criterion will also form the cornerstone of the algorithm to compute Gröbner basis, namely the *Buchberger's algorithm* (Algorithm B.2).

Theorem B.8 (Buchberger's Criterion)**.** *Let $G = \{g_1, \ldots, g_t\}$ be a basis of an ideal $I \subseteq k[x_1, \ldots, x_n]$. The basis G is a Gröbner basis of I if and only if for all pairs $i \neq j$, the remainder on division of $\mathsf{S}_\prec(g_i, g_j)$ by G is zero.*

Algorithm B.2: Buchberger$(B, \prec)$

```
Data:
    • $B = \{f_1, \ldots, f_s\} \subseteq k[x_1, \ldots, x_n]$
    • a term order $\prec$ in $k[x_1, \ldots, x_n]$
Result: $\mathcal{G}_\prec(\langle B \rangle)$
G ← B ;
repeat
    $G_{\text{old}} \leftarrow G$ ;
    for each pair $f_1, f_2 \in G$ such that, $f_1 \neq f_2$ do
        $r \leftarrow \overline{\mathsf{S}_\prec(f_1, f_2)}^G$ ;
        if $r \neq 0$ then
            $G \leftarrow G \bigcup \{r\}$ ;
        end
    end
until $G = G_{old}$;
return $G$ ;
```

B.2.4 Gröbner basis in action

From computational perspective, Gröbner basis is a very versatile structure. In this section, we will quote a few results highlighting this fact. This results also has a bearing on the discussion that is to follow in the subsequent chapters of the thesis.

Theorem B.9 (The Elimination Theorem). *Let $I \subseteq k[x_1, \ldots, x_n]$ be an ideal and let G be a Gröbner basis of I with respect to lex order, $\succ_{lex}$ where $x_1 \succ_{lex} x_2 \succ_{lex} \cdots \succ_{lex} x_n$. Then for every $0 \leq l \leq n$, the set*

$$G_l = G \bigcap k[x_{l+1}, \ldots, x_n]$$

is a Gröbner basis of the $I \bigcap k[x_{l+1}, \ldots, x_n]$. This is called the l^{th} elimination ideal *I_l of I.*

Theorem B.10 (Ideal Membership). *Let G be a Gröbner basis of $I \subseteq k[x_1, \ldots, x_n]$, $f \in k[x_1, \ldots, x_n]$. Then, $f \in I$ if and only if $\overline{f}^G = 0$.*

Theorem B.11 (Ideal Intersection). *Let $I = \langle\ f_1, \ldots, f_s\ \rangle$ and $J = \langle\ g_1, \ldots, g_t\ \rangle$ be ideals in $k[x_1, \ldots, x_n]$. Then,*

$$I \bigcap J = \langle\ yf_1, \ldots, yf_s, (1-y)g_1, \ldots, (1-y)g_t\ \rangle \bigcap k[x_1, \ldots, x_n].$$

Theorem B.12 (Ideal Saturation). *Let $I = \langle\ f_1, \ldots, f_s\ \rangle$ be an ideal in $k[x_1, \ldots, x_n]$. Then,*

$$I : x_i^\infty = \langle\ f_1, \ldots, f_s, 1 - x_i y\ \rangle \bigcap k[x_1, \ldots, x_n].$$

References

[AL94] W. W. Adams and P. Loustaunau. *An Introduction to Gröbner Bases*, volume 3 of *Graduate Studies in Mathematics*. American Mathematical Society, Rhode Island, 1994. v, 2, 89, 94

[BSR99] Anna Maria Bigatti, Robertola Scala, and Lorenzo Robbiano. Computing toric ideals. *J. Symb. Comput.*, 27(4):351–365, 1999. vi, 1, 3, 6, 33

[BU95] Fausto Di Biase and Rüdiger Urbanke. An algorithm to calculate the kernel of certain polynomial ring homomorphisms. *Experimental Mathematics*, 4:227–234, 1995. 5, 6

[Buc76] B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *SIGSAM Bull.*, 10(3):19–29, 1976. 2, 89

[CC82] Tsu-Wu J. Chou and George E. Collins. Algorithms for the solution of systems of linear diophantine equations. *SIAM J. Comput.*, 11:687–708, 1982. 5, 11

[CLO07] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. v, 2, 7, 15, 19, 23, 63, 85

[CT91] Pasqualina Conti and Carlo Traverso. Buchberger algorithm and integer programming. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 130–139, 1991. vi, 1, 3, 11

[DSS09] Mathias Drton, Bernd Sturmfels, and Seth Sullivant. Conditional Independence Lectures on Algebraic Statistics. In *Lectures on Algebraic Statistics*, volume 39 of *Oberwolfach Seminars*, chapter 3, pages 61–88. Birkhäuser Basel, Basel, 2009. 2

[Eis95] David Eisenbud. *Commutative Algebra with a View toward Algebraic Geometry*. Springer Verlag, New York, 1995. 36, 63, 85

[ES96] David Eisenbud and Bernd Sturmfels. Binomial ideals. *Duke Mathematical Journal*, 84(1):1–45, 1996. 1, 2, 4, 7, 78, 80, 81

[Fau99] J.-C. Faugre. A new efficient algorithm for computing grbner bases (f4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999. 2, 94

[Fau02] J.-C. Faugre. A new efficient algorithm for computing grbner bases without reduction to zero (f5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM. 2, 94

[Ful93] William Fulton. *Introduction to toric varieties*, volume 131 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 1993. 1

[Gil84] Robert Gilmer. *Commutative semigroup rings*. University of Chicago Press, Chicago, Illinois, 1984. 1

[GMS06] Dan Geiger, Christopher Meek, and Bernd Sturmfels. On the toric algebra of graphical models. *The Annals of Statistics*, 34(3):1463–1492, 2006. 2

[HM09] Raymond Hemmecke and Peter N. Malkin. Computing generating sets of lattice ideals and markov bases of lattices. *Journal of Symbolic Computation*, 44(10):1463–1476, 2009. v, 3, 6, 38, 58, 59

[HS95] Serkan Hosten and Bernd Sturmfels. Grin: An implementation of Gröbner bases for integer programming. *Integer Programming and Combinatorial Optimization*, 1995. vi, 1, 3, 6, 11

[Kah10] Thomas Kahle. Decompositions of binomial ideals. *Annals of the Institute of Statistical Mathematics*, 62:727–745, 2010. 10.1007/s10463-010-0290-9. 1

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8:499–507, 1979. 5, 11

[KM09] Deepanjan Kesh and Shashank K. Mehta. Generalized reduction to compute toric ideals. In *ISAAC*, pages 483–492, 2009. 11

[KM10] Deepanjan Kesh and Shashank K Mehta. Generalized reduction to compute toric ideals. *Discrete Mathematics, Algorithms and Applications (DMAA)*, 2:45–59, 2010. 11

[KM11a] Deepanjan Kesh and Shashank K Mehta. A saturation algorithm for homogeneous binomial ideals. *ISSAC 2011 poster abstract in ACM Communications in Computer Algebra*, 45(2):121–122, 2011. 35

[KM11b] Deepanjan Kesh and Shashank K. Mehta. A saturation algorithm for homogeneous binomial ideals. In *COCOA*, pages 357–371, 2011. 35

[KM12] Deepanjan Kesh and Shashank K Mehta. A divide and conquer method to compute binomial ideals. Submitted. Available at http://www.cse.iitk.ac.in/users/deepkesh/downloads/issac.pdf, 2012. 61

[MM82] Ernst W Mayr and Albert R Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3):305 – 329, 1982. v, 2

[Stu95] Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, December 1995. 1, 2, 5, 6, 7, 11, 15, 33, 36, 38, 47, 50, 57, 58

[Swa11] Irena Swanson. Expanded lectures on binomial ideals. http://people.reed.edu/~iswanson/MSRI11SwansonESbinom.pdf, 2011. 81

[Tho95] Rekha R. Thomas. A geometric buchberger algorithm for integer programming. *Mathematics of Operations Research*, 20:864–884, 1995. 1, 4

[TTN95] S. R. Tayur, R. R. Thomas, and N. R. Natraj. An algebraic geometry algorithm for scheduling in the presence of setups and correlated demands. *Mathematical Programming*, 69(3):369–401, 1995. 2, 11

[TW97] R. Thomas and R. Weismantel. Truncated gröbner bases for integer programming. *Applicable Algebra in Engineering, Communication and Computing*, 8(4):241–256, 4 1997. 2, 11

[UWZ97a] Regina Urbaniak, Robert Weismantel, and Günter M. Ziegler. A variant of the buchberger algorithm for integer programming. *SIAM J. Discrete Math.*, 10(1):96–108, 1997. 2

[UWZ97b] Regina Urbaniak, Robert Weismantel, and Günter M. Ziegler. A variant of the Buchberger algorithm for integer programming. *SIAM J. Discret. Math.*, 10(1):96–108, 1997. 11