Applications of general linear programming methods to monomial orderings in G-algebras

September 28, 2015

An der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Rheinisch–Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Master–Mathematik
angefertigte

Masterarbeit

vorgelegt von
Kai Malte Klegraf

Erstgutachterin:	Prof. Dr. Eva Zerz
Zweitgutachterin:	Prof. Dr. Gabriele Nebe
Betreuer:	Dr. Viktor Levandovskyy

Contents

Introduction

The goal of this thesis is the application of general linear and integer programming methods to compute specific admissible weighted monomial orderings in G-algebras. More precisely, we apply an algorithm to the relations of a non-commutative algebra designed to compute *balanced* weight vectors, where the notion of balance will be specified as this is new to the literature in this context. The work is organized as follows.
In the first section the basic theory of Gröbner bases in the commutative case is recalled. We will begin section 2 by considering an intersection of the mathematical areas of linear and integer programming problems and commutative algebra in terms of Gröbner bases to ease into this topic. By combining these we focus on an interesting application of Gröbner bases as a pure algebraic method to solve integer programming problems.
Afterwards, in section 3, we will lay the foundation for our computations later on by investigating certain non-commutative algebras, so called *G-algebras*. One can think of a G-algebra as a generalization of the commutative polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ over a commutative field $\mathbb{K}$. In general however, we regard a non-commutative $\mathbb{K}$-algebra

$$A = \mathbb{K}\langle x_1, \ldots, x_n \mid \{x_j x_i = c_{ij} x_i x_j + d_{ij}\}_{1 \leq i < j \leq n}\rangle,$$

with respect to a monomial ordering $\prec$. Here the coefficients c_{ij} are elements of $\mathbb{K}$ and the d_{ij} are polynomials in A. For A to be a *G-algebra*, it resp. the monomial ordering needs to fulfill certain conditions which concern the relations $x_j x_i = c_{ij} x_i x_j + d_{ij}$. It turns out that we can not only check algorithmically via linear programming methods whether these conditions are fulfilled or not, but even compute an (admissible weighted) monomial ordering that matches part of the conditions.
Furthermore we will shortly present two different filtrations of G-algebras in section 4. Again weight vectors will play an important part and hence this topic yields a good application for our results at the end of this work.
Moreover, in the utilization of our algorithm for balanced weight vectors, we will mainly focus on a specific G-algebra, namely the *(quantum) universal enveloping algebra* of a special linear Lie algebra. Hence section 5 is devoted to the corresponding theoretical background. Here we will start with the basic structure of a Lie algebra $\mathfrak{g}$, but as Lie algebras are neither commutative nor associative in general, we will embed it in its associative universal enveloping algebra $U(\mathfrak{g})$. Additionally, involving a complex parameter q to the non-commutative relations will then lead us to the so called quantum universal enveloping algebra $U_q(\mathfrak{g})$.
In section 6 we will begin our computations. As a start and also a convenient way to get a feeling for the structure and the great variety of G-algebras we will investigate numerous examples in the context of graded G-algebras. We aim to compute weights for

the generating variables $x_1, \dots, x_n$ such that the corresponding G-algebra A is graded. From the relations of A we will deduce a grading pattern by solving a general linear system of equations. Additionally we will investigate the approximation of a grading for a (commutative) polynomial in $\mathbb{K}[x_1, \dots, x_n]$. In the case that only the trivial grading, that is all weights are zero, is possible, we will present an algorithm that computes all possible highest graded parts of the polynomial that have maximal length (in number of terms) among all other gradings.
As stated in section 3, we can compute (admissible weighted) monomial orderings for G-algebras via linear programming methods. In section 7 we will finally discuss an algorithm which is not only capable of this computation, but can be used to search for solutions of a certain structure. More precisely we modify an algorithm designed to compute alternative optimal solutions of an integer programming problem to give out *balanced* weight vectors which induce admissible weighted monomial orderings. The solutions in this work tend to have a block-structure, that is we actually focus on finding block-orderings. To be more specific, the meaning of balance will be a combination of symmetry of certain blocks and specific boundaries. Although the idea for this thesis arose from the relations of a G-algebra, we kept the theory as general as possible to make it applicable within a more general context.

Acknowledgments

First of all, I am truly grateful to my advisor Dr. Viktor Levandovskyy, whose support and guidance encouraged me writing on this topic. Furthermore, his constant suggestions and remarks were very helpful throughout the working period on this thesis.
Moreover I would like to express my deepest appreciation to him and Prof. Dr. Eva Zerz for their patience and the interesting as well as fruitful discussions which brought me forward not only in this work but also in the past years.
Additionally I would like to thank Dr. Daniel Andres, whose helpfulness I could always rely on.

Last but not least I am very thankful for the constant support of my family and friends and their valuable hints and comments on this thesis.

§ 1 Basics of Gröbner Bases

We will start this work by a short introduction of Gröbner bases. As we will outline the basics of the theory of Gröbner bases, we begin with monomial orderings and essential notations, continue with different characterizations of Gröbner bases and present Buchberger's algorithm. Finally we will briefly introduce reduced Gröbner bases.
In the following section we will apply this theory to linear resp. integer programming problems.

For this section only we fix the commutative ring $R := \mathbb{K}[x_1, \ldots, x_n]$, where $\mathbb{K}$ is an arbitrary commutative field. The set of of all monomials of R, i.e.

$$Mon_n(R) := \{x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} \mid \alpha_i \in \mathbb{N},\ \text{for } 1 \leq i \leq n\}$$

will be of special interest. Furthermore, throughout this section we may also use the multi-index notation, that is instead of referring to an element $m \in Mon_n(R)$ as
$m = x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}$ we will shortly write $m = x^\alpha$, where $\alpha \in \mathbb{N}^n$.
A vital aspect when speaking of Gröbner bases are monomial orderings. While there exists a natural ordering in polynomial rings of one variable, i.e. $n = 1$, it is very important to distinguish between the different monomial orderings when $n > 1$, as this affects every step in the computations.

(1.1) Definition
We call a total ordering $\prec$ on $Mon_n(R)$ a monomial ordering if it fulfills the following conditions

1. $\prec$ is a well-ordering on $Mon_n(R)$, that is any non-empty subset of $Mon_n(R)$ has a least element with respect to $\prec$.
2. $x^\alpha \prec x^\beta$ implies $x^{\alpha+\gamma} \prec x^{\beta+\gamma}$ for all $\alpha, \beta, \gamma \in \mathbb{N}_0^n$.

(1.2) Examples
Popular examples are the *lexicographical order* $\prec_{lex}$ or the *graded lexicographical order* $\prec_{glex}$ which are defined as follows. Consider the monomials $x^\alpha, x^\beta \in Mon_n(R)$.

1. Then

$$\begin{aligned} x^\alpha \prec_{lex} x^\beta \quad &\Leftrightarrow \quad \exists\ 1 \leq i \leq n : \alpha_i < \beta_i \text{ and } \alpha_j = \beta_j\ \forall\ 1 \leq j < i. \\ &\Leftrightarrow \quad \text{The leftmost nonzero entry of } \alpha - \beta \text{ is negative.} \end{aligned}$$

2. For a monomial $f := x^\alpha \in R$ define the total degree $tdeg(f)$ of f to be $tdeg(f) := \alpha_1 + \ldots + \alpha_n$. Then we set

$$x^\alpha \prec_{glex} x^\beta \;\Leftrightarrow\; \begin{cases} \quad tdeg(x^\alpha) < tdeg(x^\beta) \\ \text{if } tdeg(x^\alpha) = tdeg(x^\beta), \text{ then } x^\alpha \prec_{lex} x^\beta. \end{cases}$$

With respect to a monomial ordering $\prec$ we conclude, that for any $f \in R \setminus \{0\}$ there exists a unique expression of f of the form

$$f = c_\alpha x^\alpha + \sum_{\beta \prec \alpha} c_\beta x^\beta,$$

where $c_\alpha, c_\beta \in \mathbb{K} \setminus \{0\}$ and $\alpha, \beta \in \mathbb{N}^n$. Note that we have only defined $\prec$ on $Mon_n(R)$ so far, but regarding all of the above we might also view $\prec$ as an ordering on $\mathbb{N}^n$.
We will use this representation of f to introduce the following notations.

(1.3) Definition
Let $\emptyset \neq F \subseteq R$ and write $f \in F$ as above.
Then we term (with respect to a given monomial ordering $\prec$)

$deg(f) := \alpha$	the degree of f,
$deg(F) := \{def(f) \mid 0 \neq f \in F\} \subseteq \mathbb{N}_0^n$,	
$lc(f) := c_\alpha$	the leading coefficient of f,
$lm(f) := c_\alpha x^\alpha$	the leading monomial of f,
$lm(F) := \{lm(f) \mid f \in F \setminus \{0\}\}$	the set of leading monomials of F,
$LM(F) := \langle lm(F) \rangle$	the ideal generated by $lm(F)$.

Moreover, it is well known that for any $f \in R$ and for any finite subset $G \subset R$ we can *divide f by G*, that is we can write

$$f = \sum_i h_i g_i + \bar{f},$$

where $h_i, \bar{f} \in R$ and $g_i \in G$ for all i. We have $lm(f) \succeq lm(h_i g_i)$ for all i and no monomial of $\bar{f}$ is contained in $LM(G)$. We call $\bar{f}$ a remainder of f with respect to G. We may also use the notation $\bar{f}^G$. Note that the remainder is not unique.

(1.4) Definition (Gröbner basis)
Let $I \subset R$ be an ideal in R. A subset $\mathcal{G} \subset R$ is called a Gröbner basis of I if

$$LM(\mathcal{G}) = LM(I).$$

Now consider $\alpha, \beta \in \mathbb{N}_0^n$. We define $\alpha \vee \beta \in \mathbb{N}_0^n$ by

$$(\alpha \vee \beta)_i := \max\{\alpha_i, \beta_i\} \text{ for } 1 \leq i \leq n.$$

Hence $\alpha_i, \beta_i \leq (\alpha \vee \beta)_i$ for all i. This notation allows us to give a simplified expression of the well known *S-polynomials*:
Take $f, g \in R \setminus \{0\}$ with $deg(f) = \alpha$ and $deg(g) = \beta$. We define the *S-polynomial* $S(f, g) \in R$ of f and g by

$$S(f, g) := lc(g)x^{\alpha\vee\beta-\alpha} f - lc(f)x^{\alpha\vee\beta-\beta} g.$$

(1.5) Example
Fix $R = \mathbb{R}[x, y, z]$ and consider the lexicographical ordering, where $x \succ y \succ z$. Regarding the monomials $f_1 := x^2y + xz, f_2 := xyz + yz$ and $f_3 := xy + y^2z + 1$ we have for example

$$\begin{array}{rcl} S(f_1, f_2) & = & xz^2 - xyz = -xyz + xz^2 \\ \text{or } S(f_2, f_3) & = & yz - y^2z^2 - z = -y^2z^2 + yz - z. \end{array}$$

Using the S-polynomials, we obtain an important characterization of Gröbner bases, see [3] by Becker and Weispfenning for example.

(1.6) Proposition (Buchberger criterion)
Let $I \subset R$ be an ideal in R and let G be a basis of I.
Then we have

$$\begin{gathered} G \text{ is a Gröbner basis of } I \\ \Leftrightarrow \\ \text{A remainder of } S(f, g) \text{ with respect to } G \text{ is zero for all } f, g \in G. \end{gathered}$$

We are now able to state the **Buchberger algorithm**:
Let F be a non-empty finite subset of $R = \mathbb{K}[x_1, \ldots, x_n] \setminus \{0\}$.

Algorithm 1 Buchberger algorithm

```
1: $i := 0$
2: $F_0 := F$
3: do
4:     $\mathcal{G} := F_i$
5:     $F_{i+1} := F_i \cup \left\{ \overline{S(f,g)}^{F_i} \mid f, g \in F_i \right\} \setminus \{0\}$
6: while $F_{i+1} \neq F_i$
7: return $\mathcal{G}$.
```

(1.7) Example
Consider Example (1.5) again and set $F := \{f_1, f_2, f_3\}$, resp. $I = \langle F \rangle$.
Using the algorithm we might compute the Gröbner basis

$$\mathcal{G} := \{x + 1, y - z, y^3 - y + 1\}.$$

In the interests of clarity we stress that in general the Gröbner basis of an ideal is not unique even if we fix a monomial ordering. To achieve a notion of uniqueness however one considers *reduced* Gröbner bases. More precisely, we call $\mathcal{G}$ a *reduced* Gröbner basis of an ideal I if the three following conditions hold:

1. All of the $f \in \mathcal{G}$ are monic, i.e. $lc(f) = 1$ for all $f \in \mathcal{G}$.
2. $\mathcal{G}$ is *minimal*, that is for all $f \in \mathcal{G}$ it holds

$$def(f) \notin deg(\mathcal{G} \setminus \{f\}) + \mathbb{N}^n.$$

3. Each $f \in \mathcal{G}$ is in *normal form modulo* I, that is for all $f - x^{deg(f)}$ it holds

$$f - x^{deg(f)} \in \bigoplus_{\alpha \notin deg(\mathcal{G}) + \mathbb{N}^n} \mathbb{K}x^\alpha.$$

While the first condition eliminates an obvious reason for the lack of uniqueness, the second one assures, that no $f \in \mathcal{G}$ is redundant. The third condition makes sure, that the leading term of any $f \in \mathcal{G}$ is not a term of any $f' \in \mathcal{G} \setminus \{f\}$. Let us take up the example from above.

(1.8) Example
Each element of $\mathcal{G} := \{x+1, y-z, y^3-y+1\}$ is clearly monic. As

$$deg(\mathcal{G}) = \{(1,0,0),(0,1,0),(0,3,0)\}$$

we conclude, that $\mathcal{G}$ is not minimal and therefore not reduced. A straightforward computation yields, that

$$\mathcal{G}' = \{x+1, y-z, z^3-z+1\}$$

is the reduced Gröbner basis of $I = \langle x^2y+xz, xyz+yz, xy+y^2z+1\rangle$ with respect to the lexicographical ordering.

While the main applications of Gröbner bases are well known, for example the *ideal membership problem*, we will focus on a different scope in the following section. We will discuss how to translate *integer programming problems* into the language of Gröbner bases and use the methods hereof to solve the given problems.

§2 Gröbner Bases and Integer Programming

Although the topic of this thesis is the application of linear resp. integer optimization methods to non-commutative algebras, we first focus on a simple link of these two different mathematical areas. This section is a pleasant way to ease into the merging of those and serves as an introduction to integer programming in combination with the theory of Gröbner bases. We have taken the work of Cox, Little and O'Shea [6] as a basis.
Additionally we assume the regarded rings in this section to be commutative.
We will start with the basics of integer programming and outline the problems to be solved. In order to adopt the theory of Gröbner bases to this setting we will translate it into the language of polynomials and monomial orderings.

Let us start with a simple two-dimensional example of an integer programming problem (shortly IP) to illustrate the typical features of this class of problems.

(2.1) Example
Consider a (very small) guitar manufacturing company producing two guitars, one electric and one acoustic. As any other company in the business it struggles to keep the economic progress going and hence is focused on maximizing of profit.
The guitars are manufactured in a factory out of town and each month they use a truck to supply their stores in a single ride. To keep the example easy we will only consider two constraints. Let the first one be given by the fact that the overall basic materials hold out for manufacturing at most 12 guitars in total per month. And second, the truck only offers 51 cubic meters, whereas an acoustic, resp. electric, guitar takes up 6, resp. 3, cubic meters due to additional equipment and packing.
Say p_e, resp. p_a, denote the number of sold electric, resp. acoustic, guitars per month, where the price is 220€ for an electric and 308€ for an acoustic guitar. The information above translated into the language of integer programming looks like follows:

$$\begin{array}{lrl} \text{Max} & 220p_e + 308p_a & \\ \text{s.t.} & p_e + \quad p_a & \leq 12 \\ & 3p_e + \quad 6p_a & \leq 51, \end{array} \tag{2.1}$$

where $p_a, p_e \in \mathbb{Z}_{\geq 0}$.

Note, that p_a and p_e must be integers. This is a key feature of *integer* programming problems. We usually want to maximize or minimize the value of a linear function ℓ, here $\ell(p_e, p_a) = 220p_e + 308p_a$, subject to a given set of constraints. We call ℓ the

objective function and $\ell(p_e, p_a)$ the *objective value* at the point (p_a, p_e). Note that the *optimal* objective value of an integer programming problem is unique, whereas there may be multiple optimal solutions yielding the same optimal objective value. The variables in the objective function, here p_e and p_a, are also called *decision variables.* Instead of maximizing the objective function one might also want to minimize it under certain conditions, for example in the context of minimizing production costs.
An abstract integer programming problem takes the form

$$\begin{array}{rrcl}
\text{Max (or Min)} & c_1p_1 & + & \dots + c_np_n \\
\text{s.t.} & a_{11}p_1 & + & \dots + a_{1n}p_n \leq (\text{ or } \geq)\ b_1 \\
 & a_{21}p_1 & + & \dots + a_{2n}p_n \leq (\text{ or } \geq)\ b_2 \\
 & & \vdots & \\
 & a_{m1}p_1 & + & \dots + a_{mn}p_n \leq (\text{ or } \geq)\ b_m \\
 & p_j & \in & \mathbb{Z}_{\geq 0},
\end{array}$$

where we assume that $c_j, a_{ij}, b_i \in \mathbb{Z}$.

As one can see in the example above, an integer programming problem can have several solutions, for example the points $(5,5)$ or $(4,6)$ also satisfy the inequalities in (2.1), yielding the objective values 2640 and 2112. This naturally raises the question if one can name the set of all possible (not necessarily optimal) solutions to the given problem. Such a solution, i.e. a point P satisfying every given constraint, is called *feasible.*

(2.2) Definition
The *feasible region* of an integer programming problem is the set $\mathcal{P}$ of all points $P = (p_1, \dots, p_n)^T \in \mathbb{R}^n$ satisfying the inequalities in the statement of the problem.

Note that $\mathcal{P}$ is allowed to be empty, i.e. in that case the problem does not have any solution. Furthermore if $\mathcal{P} \cap \mathbb{Z}^n = \emptyset$ we say, that the problem does not have an *integer* solution. If a point P fails to satisfy the constraints it is said to be *infeasible.*

A pleasant property of two-dimensional IP's is that they are easy to visualize. The feasible region of (2.1) is given by

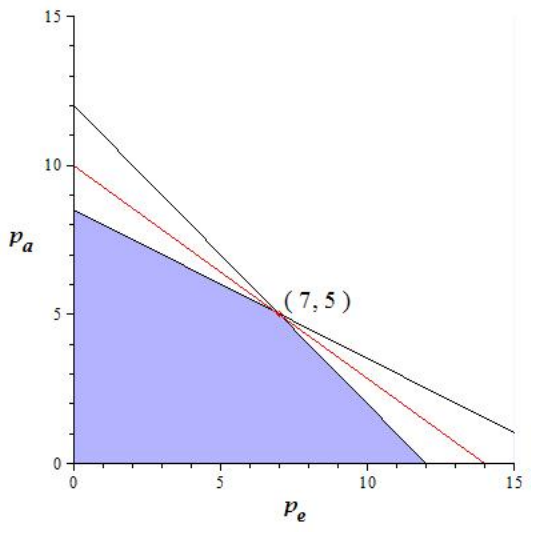

Figure 1: Feasible region of (2.1)

The red line indicates the objective function and the point at position $(7,5)$ marks the optimal solution. That this point is in fact the optimal solution will be verified below using Gröbner bases. Note that, in general, the optimal solution of an *integer* programming problem may *not* be one of the vertices of the feasible region, this is a well-known fact for *linear* programming problems. Moreover it is easy to construct a feasible region where no vertex is an element of $\mathbb{Z}^n$.

In the discussion so far we have considered integer minimization and maximization problems subject to inequalities of the form $\leq$ and/or $\geq$. This was merely useful for introductory purposes. As the following three observations will show, focussing on minimization problems subject to *equalities* is sufficient to cover all problems of this class.

First, we only need to consider the problem of minimizing the linear function ℓ, since maximizing ℓ on a set of integer n-tuples is equivalent to minimizing the function $-\ell$.

Similar to the first step we can replace each inequality of the form

$$a_{i1}p_1 + \ldots + a_{in}p_n \geq b_i$$

by the equivalent form

$$-a_{i1}p_1 - \ldots - a_{in}p_n \leq -b_i.$$

And finally, we can rewrite each inequality as an equality by introducing a set of new variables, called *slack variables.* For example, an inequality

$$a_{i1}p_1 + \ldots + a_{in}p_n \leq b_i$$

can be replaced by an equality by introducing a new variable p_{n+1}:

$$a_{i1}p_1 + \ldots + a_{in}p_n + p_{n+1} = b_i,$$

that is, p_{n+1} “takes up the slack”. These variables also appear in the objective function however with coefficient zero.
Hence each integer programming problem can be transformed into its equivalent, the so called *standard form* :

$$(*) \quad \begin{array}{lrclcrcl} \text{Min} & c_1p_1 & + & \ldots & + c_np_n & & \\ \text{s.t.} & a_{11}p_1 & + & \ldots & + a_{1n}p_n & = & b_1 \\ & a_{21}p_1 & + & \ldots & + a_{2n}p_n & = & b_2 \\ & & \vdots & & & & \\ & a_{m1}p_1 & + & \ldots & + a_{mn}p_n & = & b_m \\ & p_j & \in & \mathbb{Z}_{\geq 0}, & j = 1, \ldots n, & & \end{array}$$

with $c_j, a_{ij}, b_i \in \mathbb{Z}$, where we assume that $(p_1, \ldots, p_n)$ already contains all slack variables. Hence from now on when speaking of an integer programming problem we will assume it is in standard form. For example, the standard form of (2.1) is

$$\begin{array}{lrrcl} \text{Min} & -220p_e - & 308p_a & & \\ \text{s.t.} & p_e + & p_a + p_3 & = & 12 \\ & 3p_e + & 6p_a + p_4 & = & 51 \end{array}$$

where $p_e, p_a, p_3, p_4 \in \mathbb{Z}_{\geq 0}$.

Having established the basics of integer programming, we will now translate the underlying problem to the language of Gröbner bases. Therefore we have to make this a question about polynomials rather than a system of linear equations.
As a start, we will make the assumption that all coefficients in the constraints are nonnegative, that is $a_{ij} \geq 0, b_i \geq 0$.
First let us introduce a new variable z_i for each equation of the standard form and exponentiate to get a new equality

$$z_i^{a_{i1}p_i + \ldots + a_{in}p_n} = z_i^{b_i}$$

for each $i = 1, \ldots, m$. Next we multiply all left hand sides and all right hand sides of these equations and rearrange the exponents to obtain the equality

$$\prod_{j=1}^{n} \left(\prod_{i=1}^{m} z_i^{a_{ij}} \right)^{p_j} = \prod_{i=1}^{m} z_i^{b_i},$$

which can be used to give a characterization of the feasible region.

(2.3) Proposition
Let $\mathbb{K}$ be a field, and define $\varphi : \mathbb{K}[w_1, \ldots, w_n] \to \mathbb{K}[z_1, \ldots, z_m]$ by setting

$$\varphi(w_j) = \prod_{i=1}^{m} z_i^{a_{ij}}$$

for each $j = 1, \ldots, n$, and $\varphi(g(w_1, \ldots, w_n)) = g(\varphi(w_1), \ldots, \varphi(w_n))$ for a general polynomial $g \in \mathbb{K}[w_1, \ldots, w_n]$. Then $(p_1, \ldots, p_n)$ is an integer point in the feasible region if and only if φ maps the monomial $w_1^{p_1} w_2^{p_2} \cdots w_n^{p_n}$ to the monomial $z_1^{b_1} \cdots z_m^{b_m}$.

Proof
Let $P = (p_1, \ldots, p_n)$ be an integer point in the feasible region. Now set

$$g(w_1, \ldots, w_n) := w_1^{p_1} w_2^{p_2} \cdots w_n^{p_n},$$

then by the assumption we have

$$\begin{aligned} \varphi(g(w_1, \ldots, w_n)) &= g(\varphi(w_1), \ldots, \varphi(w_n)) \\ &= g\left(\prod_{i=1}^{m} z_i^{a_{i1}}, \ldots, \prod_{i=1}^{m} z_i^{a_{in}} \right) \\ &= \left(\prod_{i=1}^{m} z_i^{a_{i1}} \right)^{p_1} \cdots \left(\prod_{i=1}^{m} z_i^{a_{in}} \right)^{p_n}. \end{aligned}$$

A rearrangement of the exponents and the fact that P lies in the feasible region, i.e. $\sum_{j=1}^{n} a_{ij} p_j = b_i$ holds for each $i = 1, \ldots, m$ yields

$$\left(\prod_{i=1}^{m} z_i^{a_{i1}} \right)^{p_1} \cdots \left(\prod_{i=1}^{m} z_i^{a_{in}} \right)^{p_n} = z_1^{b_1} \cdots z_m^{b_m}.$$

For the other direction let g be defined as above and let $\varphi(w_1^{p_1} \cdots w_n^{p_n}) = z_1^{b_1} \cdots z_m^{b_m}$. Now the assertion follows by reading the equations above in another way, namely

$$z_1^{b_1} \cdots z_m^{b_m} = \varphi(w_1^{p_1} \cdots w_n^{p_n}) = \ldots = \left(\prod_{i=1}^{m} z_i^{a_{i1}} \right)^{p_1} \cdots \left(\prod_{i=1}^{m} z_i^{a_{in}} \right)^{p_n},$$

hence $\sum_{j=1}^{n} a_{ij} p_j = b_i$ has to hold for each $i = 1, \ldots, m$, that is P is a point in the feasible region. $\square$

Let us take up our example. Then,

$$\begin{array}{rcl} \varphi : \mathbb{K}[w_1, w_2, w_3, w_4] & \to & \mathbb{K}[z_1, z_2] \\ w_1 & \mapsto & z_1 z_2^3 \\ w_2 & \mapsto & z_1 z_2^6 \\ w_3 & \mapsto & z_1 \\ w_4 & \mapsto & z_2. \end{array}$$

A point $P = (p_e, p_a, p_3, p_4)$ lies in the feasible region, if

$$\varphi(w_1^{p_e}, w_2^{p_a}, w_3^{p_3}, w_4^{p_4}) = z_1^{12} z_2^{51}.$$

In general φ may not be surjective. So, to check if a given monomial in $\mathbb{K}[z_1, \dots, z_m]$ is in the image of φ, i.e. if the corresponding integer program is feasible, we have to take a closer look at φ.
The transformation rule in (2.3) tells us that the image of φ in $\mathbb{K}[z_1, \dots, z_m]$ is generated by the $f_j = \prod_{i=1}^m z_i^{a_{ij}}$, hence we can write

$$im(\varphi) = \mathbb{K}[f_1, \dots, f_n] \subset \mathbb{K}[z_1, \dots, z_m].$$

Now the test for membership in $im(\varphi)$ is given by the following proposition.

(2.4) Proposition
Suppose that $f_1, \dots, f_n \in \mathbb{K}[z_1, \dots, z_m]$ are given. Fix a monomial order in $\mathbb{K}[z_1, \dots, z_m, w_1, \dots, w_n]$ with the *elimination property*: any monomial containing one of the z_i is greater than any monomial containing only the w_j. Let $\mathcal{G}$ be a Gröbner basis (GB) for the ideal

$$I = \langle f_1 - w_1, \dots, f_n - w_n \rangle \subset \mathbb{K}[z_1, \dots, z_m, w_1, \dots, w_n]$$

and for each $f \in \mathbb{K}[z_1, \dots, z_m]$, let $\bar{f}^{\mathcal{G}}$ be the remainder of the division of f by $\mathcal{G}$.
Then

a) A polynomial f satisfies $f \in \mathbb{K}[f_1, \dots, f_n]$ if and only if $g = \bar{f}^{\mathcal{G}} \in \mathbb{K}[w_1, \dots, w_n]$.

b) If $f \in \mathbb{K}[f_1, \dots, f_n]$ and $g = \bar{f}^{\mathcal{G}} \in \mathbb{K}[w_1, \dots w_n]$ as in part a), then $f = g(f_1, \dots, f_n)$ giving an expression for f as a polynomial in the f_j.

c) If each f_j and f are monomials and $f \in \mathbb{K}[f_1, \dots, f_n]$, then g is also a monomial.

Proof
For a proof of parts a) and b), see [6] Proposition 7 of Chapter 7, §3.
For the proof of part c), first note that each generator of I is a difference of two monomials. Applying Buchberger's algorithm to compute a Gröbner base $\mathcal{G}$ of I yields that each S-polynomial is again a difference of two monomials. This is due to the fact that an S-polynomial of two generators of I is a subtraction of two differences of two monomials, where the leading terms cancel. The same holds for the remainder calculation. Hence each element of $\mathcal{G}$ is a difference of two monomials. Furthermore, by a similar argument the remainder of a division of a *monomial* by $\mathcal{G}$ is a again a monomial. Hence by the assumption in part c), that is f_j and f are monomials with $f \in \mathbb{K}[f_1, \dots, f_n]$ and taking parts a) and b) into account, i.e. $g = \bar{f}^{\mathcal{G}} \in \mathbb{K}[w_1 \dots, w_n]$, the discussion above implies that g must be a monomial. □

An alternative way to state part c) is that in the situation of Proposition (2.3), each monomial in $im(\varphi)$ is the image of some monomial in $\mathbb{K}[w_1, \dots, w_n]$.

Consider the example (2.1) again, then

$$I = \langle z_1 z_2^3 - w_1, z_1 z_2^6 - w_2, z_1 - w_3, z_2 - w_4 \rangle.$$

We use the lexicographical ordering with

$$z_1 \succ z_2 \succ w_4 \succ w_3 \succ w_2 \succ w_1$$

to eliminate terms involving slack variables if possible and apply `Singular` [9] to compute the Gröbner basis $\mathcal{G}$:

$$\begin{aligned} g_1 &:= z_1 - w_3, \\ g_2 &:= z_2 - w_4, \\ g_3 &:= w_4^3 w_3 - w_1, \\ g_4 &:= w_4^3 w_1 - w_2, \\ g_5 &:= w_3 w_2 - w_1^2. \end{aligned}$$

Now consider $f = z_1^{12} z_2^{51}$. We can use g_1 and g_2 to reduce f to $w_3^{12} w_4^{51}$ to see, that f is in the the image of φ. A further reduction with respect to $\mathcal{G}$ yields

$$\bar{f}^{\mathcal{G}} = w_2^5 w_1^7$$

which corresponds to $(p_e, p_a, p_3, p_4) = (7, 5, 0, 0)$, that is the optimal solution. This should surprise us as we have not taken the objective function into consideration. However the following will clarify this “accident”.

Since we have only dealt with the feasibility of an integer program in the context of Gröbner bases so far, we left out the objective function until this point. Taking the objective function into account is usually linked to using a monomial order designed specifically for the problem at hand.

(2.5) Definition
A monomial order $\prec$ on $\mathbb{K}[z_1, \dots, z_m, w_1, \dots, w_n]$ is said to be *adapted* to an integer programming problem $(*)$ if it has the following two properties:

1. Elimination:
   Any monomial containing one of the $z_i' s$ is greater than any monomial containing only the w_j.

2. Compatibility with ℓ :
   Let $P = (p_1, \dots, p_n)$ and $P' = (p_1', \dots, p_n')$. If the monomials $w^P, w^{P'}$ satisfy $\varphi(w^P) = \varphi(w^{P'})$ and $\ell(p_1, \dots, p_n) > \ell(p_1', \dots, p_n')$, then $w^p \succ w^{p'}$.

(2.6) Theorem
Consider an integer programming problem in standard form $(*)$. Assume all $a_{ij}, b_i \geq 0$ and let $f_j = \prod_{i=1}^m z_i^{a_{ij}}$ be as before. Let $\mathcal{G}$ be a Gröbner basis for

$$I = \langle f_1 - w_1, \dots, f_n - w_n \rangle \subset \mathbb{K}[z_1, \dots, z_m, w_1, \dots, w_n]$$

with respect to any adapted monomial order. Then if $f = z_1^{b_1} \cdots z_m^{b_m}$ is in $\mathbb{K}[f_1, \dots, f_n]$, the remainder $\bar{f}^{\mathcal{G}} \in \mathbb{K}[w_1, \dots, w_n]$ will give a solution of $(*)$ minimizing ℓ.

Proof
Let $\mathcal{G}$ be a Gröbner basis for I with respect to an adapted monomial order. Assume to the contrary that $P = (p_1, \dots, p_n)$ is not a minimum of ℓ, but $\varphi(w_P) = f$ and $w^P = \bar{f}^{\mathcal{G}}$. Hence there is some $P' = (p_1', \dots, p_n') \neq P$ such that $\varphi(w^{P'}) = f$ and $\ell(P') < \ell(P)$.
Now for $0 \neq h = w^P - w^{P'}$ we have $\varphi(h) = f - f = 0$. According to the following lemma this implies $h \in I$, but then h must reduce to zero under $\mathcal{G}$. However, according to the definition of an adapted order $w^P \succ w^{P'}$ and hence w^P is the leading term of h and already reduced with respect to $\mathcal{G}$. This is a contradiction and therefore P is a minimum of ℓ. □

(2.7) Lemma
Let $f_i \in \mathbb{K}[z_1, \ldots, z_m],\ i = 1, \ldots, n$ as above and recall the mapping

$$\begin{array}{rcl} \varphi : \mathbb{K}[w_1, \ldots, w_n] & \to & \mathbb{K}[z_1, \ldots, z_m] \\ w_i & \mapsto & f_i \end{array}$$

of (2.3). Let $I = \langle f_1 - w_1, \ldots, f_n - w_n \rangle \subset \mathbb{K}[z_1, \ldots, z_m, w_1, \ldots, w_n]$.
Then if $h \in \mathbb{K}[w_1, \ldots, w_n]$ satisfies $\varphi(h) = 0$, then $h \in I \cap \mathbb{K}[w_1, \ldots, w_n]$.

Proof
Let $h \in \mathbb{K}[w_1, \ldots, w_n]$ and replace each w_i in h by $w_i = f_i - (f_i - w_i)$, that is we consider h as a polynomial in $\mathbb{K}[z_1, \ldots, z_m, w_1, \ldots, w_n]$. If we expand h after the replacement, then we obtain a new form of h:

$$h = h_f + t_1(f_1 - w_1) + \ldots + t_n(f_n - w_n),$$

where $h_f, t_i \in \mathbb{K}[f_1, \ldots, f_n]$ for $i = 1, \ldots, n$. The crucial observation is that h_f is the same polynomial as h only with each w_i replaced by f_i. Now expand the domain of φ to $\mathbb{K}[z_1, \ldots, z_m, w_1, \ldots, w_n]$ by setting $\varphi(z_j) = z_j$ for each $j = 1, \ldots, m$. In particular $\varphi(f_i) = f_i$ for each $i = 1, \ldots n$. Now $\varphi(h) = 0$ yields

$$0 = \varphi(h) = \varphi(h_f + t_1(f_1 - w_1) + \ldots + t_n(f_n - w_n)) = h_f.$$

Replacing each f_i in h_f again by $f_i = w_i + (f_i - w_i)$ and taking the observation above into account we obtain

$$0 = h_f = h + \tilde{t}_1(f_1 - w_1) + \ldots + \tilde{t}_n(f_n - w_n),$$

where $\tilde{t}_i \in \mathbb{K}[w_1, \ldots, w_n]$ for each $i = 1, \ldots, n$.
Hence

$$h = -(\tilde{t}_1(f_1 - w_1) + \ldots + \tilde{t}_n(f_n - w_n)) \in I$$

which proves the assertion. $\square$

A closer look at Theorem (2.6) reveals a GB-algorithm for solving integer programming problems with all $a_{ij}, b_i \geq 0$:

Algorithm 2 GB-Algorithm

```
    Input:  $A$, $b$  from  $(*)$  and an adapted monomial order  $\prec$
    Output : a solution of  $(*)$,  if one exists.
 1: $f_j := \prod_{i=1}^{m} z_i^{c_{ij}}$
 2: $I := \langle f_1 - w_1, \ldots, f_n - w_n \rangle$
 3: $\mathcal{G} :=$ Gröbner basis of I with respect to $\prec$
 4: $f := \prod_{i=1}^{m} z_i^{b_i}$
 5: $g := \bar{f}^{\mathcal{G}}$
 6: if $g \in \mathbb{K}[w_1, \ldots, w_n]$ then
 7:     its exponent vector gives a solution
 8: else
 9:     there is no solution
10: end if
```

Now in [6] the authors mention that there are cases where the minimum is not unique and if so the method will only find one minimum. This is a usual concern with given algorithms for integer programming problems. We will investigate this issue along the discussion of *balancing weights* in section 7.

Note that we have only dealt with a special case of integer programming problems so far as we assumed the coefficients a_{ij} and b_i to be non-negative. What changes if we drop these restrictions on a_{ij} and b_i ?
On one side, it is easy to see that there are no changes in the interpretation of the integer programs. However since negative coefficients a_{ij} lead to negative exponents in the corresponding polynomials, it is also obvious, that the theory of Gröbner bases above is not directly applicable in this case. Instead of starting again from the top one can fix this problem by considering the well known *Laurent polynomials*, see [6], chapter 8.

§ 3 Non-commutative Algebras

As a student, one might prefer to work on commutative rings or algebras due to arising "difficulties" which come along with the lack of commutativity. However, at a closer look the study of non-commutative rings is more natural as for an arbitrary ring its chance to be non-commutative is quite great. For example, each commutative ring R yields a list of non-commutative rings, for instance the matrix ring $M_n(R)$ where $n \in \mathbb{N}$, which plays a very important part even in the basics of linear algebra. Further prominent examples are the division ring of quaternions $\mathbb{H}$ by Hamilton or any *free* $\mathbb{K}$*-ring* along with rings, resp. algebras, with relations which will be presented in more detail below. An indicator of the "size" of commutativity for a given arbitrary ring R is the *center* of the ring $Z(R)$. It consists exactly of those elements which commute with all elements of R, that is $Z(R) = \{a \in R \mid ar = ra \ \forall r \in R\}$. In the case when R is commutative it obviously holds $R = Z(R)$. Considering the free $\mathbb{K}$-ring $R = \mathbb{K}\langle x, y\rangle$ for example, one concludes $Z(R) = \mathbb{K}$ and hence view R as a $\mathbb{K}$-algebra.
Examining non-commutative rings, resp. algebras, in detail would go beyond the scope of this work and therefore we will outline important facts when necessary. In view of Gröbner bases however, we wish to stress at this point the importance to distinguish between *left*, *right* and *two-sided* ideals or for that matter the division of an element by another. A detailed treatment of non-commutative Gröbner bases is given by Levandovskyy in [15]. It is also, along with [1] by Andres, a basis of the following subsections.

— *G-algebras* —

In this section we will investigate non-commutative algebras, especially *G-algebras.* As a start consider the free monoid Mon_n generated by the indeterminates $x_1, \dots, x_n$, i.e.

$$Mon_n := \{x_{i_1} \cdot \ldots \cdot x_{i_k} \mid k \in \mathbb{N}_0 \text{ and } 1 \leq i_j \leq n \text{ for } 1 \leq j \leq k\}.$$

We will call the elements of Mon_n *words* or *monomials.* If we take the set of all finite linear combinations of the elements of Mon_n over a field $\mathbb{K}$, we obtain the *free associative* $\mathbb{K} - algebra$

$$F_n := \mathbb{K}\langle x_1, \dots, x_n\rangle = span_{\mathbb{K}}(Mon_n)$$

with concatenation as multiplication. The *empty word* is given by $1 \in \mathbb{K}$.
Considering the elements of Mon_n, resp. F_n, is usually linked to the need for a corresponding ordering $\prec$.

(3.1) Definition
We call a total ordering $\prec$ on Mon_n a *well-ordering*, if each non-empty subset of Mon_n contains a least element with respect to $\prec$.

(3.2) Definition
We call a total ordering $\prec$ on Mon_n a *monomial ordering*, if the following conditions are satisfied:

i) $\prec$ is a well-ordering.

ii) For all $m_1, m_2 \in Mon_n$ it holds:
$$m_1 \prec m_2 \text{ implies } p_1 \cdot m_1 \cdot p_2 \prec p_1 \cdot m_2 \cdot p_2 \quad \forall\, p_1, p_2 \in Mon_n.$$

iii) For all $m_1, m_2 \in Mon_n$ it holds:
$$m_1 = p_1 \cdot m_2 \cdot p_2 \neq m_2 \text{ with } p_1, p_2 \in Mon_n \text{ implies } m_2 \prec m_1.$$

(3.3) Examples

1. The *lexicographical* ordering $\prec_{lp}$ for example is a monomial ordering, which is defined by
$$x^\alpha \prec_{lp} x^\beta \Leftrightarrow\ \exists\, 1 \le i \le n : \alpha_i < \beta_i \text{ and } \alpha_j = \beta_j\ \forall\, 1 \le j < i.$$

2. The *degree reverse lexicographical ordering* $\prec_{dp}$ is defined as follows:
We say $x^\alpha \prec_{dp} x^\beta$ if and only if $\sum_{i=1}^n \alpha_i < \sum_{i=1}^n \beta_i$ or $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ and the last non-zero entry of $\beta - \alpha$ is negative.

3. Additionally to these "standard" monomial orderings it is very common and particularly of interest for this work to consider *weighted* monomial orderings. For this one combines a monomial ordering $\prec$ and a fixed, so called *weight vector* $\omega \in \mathbb{R}^n_{\geq 0}$. The resulting weighted monomial ordering $\prec_\omega$ is defined by
$$x^\alpha \prec_\omega x^\beta \Leftrightarrow \begin{cases} \quad \sum_{i=1}^n \omega_i\alpha_i < \sum_{i=1}^n \omega_i\beta_i \\ \text{or } \sum_{i=1}^n \omega_i\alpha_i = \sum_{i=1}^n \omega_i\beta_i \text{ and } x^\alpha \prec x^\beta. \end{cases}$$

With respect to a monomial ordering $\prec$ on Mon_n any $f \in F_n \setminus \{0\}$ can be uniquely written as
$$f = c \cdot m + f'$$
with $c \in \mathbb{K} \setminus \{0\}$, $m \in Mon_n$ and $f' \in F_n$, where f' consists of monomials $m' \in Mon_n$ with $m' \prec m$.

Hence we can introduce the following notations:

1. $lm(f) := m$ is termed the *leading monomial* of f.
2. $lc(f) := c$ is termed the *leading coefficient* of f.

(3.4) Remark
In view of the following definition of a G-algebra we recall the well-known fact, that any finitely generated associative $\mathbb{K}$-algebra A can be viewed as a quotient of F_n by a two-sided Ideal I, that is $A \cong F_n/I$. The ideal I contains information about the given relations of A (examples follow below) and is hence termed the *ideal of relations of A*. Furthermore we proceed on the assumption that I is finitely generated.

Now it is time to introduce the notion of a G-algebra.

(3.5) Definition
Let A be an associative $\mathbb{K}$-algebra in n indeterminates $x_1, \ldots, x_n$ with respect to the relations

$$x_j x_i = c_{ij} \cdot x_i x_j + d_{ij} \qquad \forall 1 \leq i \lneq j \leq n \tag{1}$$

where $c_{ij} \in \mathbb{K} \setminus \{0\}$ and $d_{ij} \in span_{\mathbb{K}}\{x_1^{\alpha_1} x_2^{\alpha_2} \cdot \ldots \cdot x_n^{\alpha_n} | \, \alpha_i \in \mathbb{N}_0, 1 \leq i \leq n\}$.
We call A a *G-algebra*, if the following two conditions hold:

a) *Ordering condition* :
There exists a monomial ordering $\prec$ on Mon_n satisfying $x_i x_j \prec x_j x_i$, such that $d_{ij} = 0$ or $lm(d_{ij}) \prec x_i x_j$ holds for all $1 \leq i \lneq j \leq n$.

b) *Non-degeneracy condition* :
The polynomial

$$NDC_{ijk} := c_{ik}c_{jk}d_{ij}x_k - x_k d_{ij} + c_{jk}x_j d_{ik} - c_{ij}d_{ik}x_j + d_{jk}x_i - c_{ij}c_{ik}x_i d_{jk}$$

vanishes on A for all $1 \leq i \lneq j \lneq k \leq n$.

To put it simply, the non-degeneracy condition assures associativity on A with respect to the given relations. More precisely, $(x_k x_j)x_i = x_k(x_j x_i)$ is equivalent to

$$\begin{aligned} 0 &= (x_k x_j)x_i - x_k(x_j x_i) \\ &= c_{ik}c_{jk}d_{ij}x_k - x_k d_{ij} + c_{jk}x_j d_{ik} - c_{ij}d_{ik}x_j + d_{jk}x_i - c_{ij}c_{ik}x_i d_{jk} \\ &= NDC_{ijk}. \end{aligned}$$

In general when speaking of a G-algebra we usually fix a monomial ordering $\prec$.
An essential part of this work is the investigation of these orderings, especially *weighted* orderings as we will see later on.

With the recall above in mind, the corresponding ideal of relations in the definition of a G-algebra is $I = \langle x_j x_i - c_{ij} \cdot x_i x_j - d_{ij} \rangle$.
Let us take a look at a few examples.

(3.6) Examples (G-algebra)

i) A trivial example is given by the commutative polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$.

ii) An easy non-commutative example is given by the so called *quasi*-commutative $\mathbb{K}$-algebra

$$\mathbb{K}\langle x_1, \ldots, x_n \mid x_j x_i = c_{ij} x_i x_j \text{ for } 1 \leq i < j \leq n \rangle,$$

where $c_{ij} \in \mathbb{K}$ for all $1 \leq i < j \leq n$.

iii) A famous example of a G-algebra is the n-th polynomial *Weyl algebra*:

$$\begin{aligned} D_n := \mathbb{K}\langle x_1, \ldots x_n, \partial_1, \ldots, \partial_n \mid \ x_j x_i &= x_i x_j, \partial_j \partial_i = \partial_i \partial_j, \\ \partial_j x_i &= x_i \partial_j + \delta_{ij} \text{ for } 1 \leq i, j \leq n \rangle. \end{aligned}$$

G-algebras were first introduced in the literature, at least under this name, by J. Apel in his dissertation [2], however without the non-degeneracy condition which is due to the work of Mora in [16]. They are also called *algebras of solvable type* by Kandri-Rody and Weispfenning in [14]. Yet another name for a G-algebra is the so called *PBW algebra*, see [5] by Bueso et al. for example, which presents a further important characterization of G-algebras via the *Poincaré-Birkhoff-Witt*-basis (shortly PBW basis):
Recall that elements of Mon_n are called monomials. Consider the subset of *standard monomials*

$$SM_n := \{x_{i_1}^{\alpha_1} \cdot x_{i_2}^{\alpha_2} \cdot \ldots \cdot x_{i_r}^{\alpha_r} \mid 1 \leq i_1 \lneq i_2 \lneq \ldots \lneq i_r \leq n,\ \alpha_k \geq 0\}.$$

If the set of standard monomials forms a $\mathbb{K}$-basis of an algebra $A = F_n/I$, then A is said to have a PBW basis (in the indeterminates $x_1, \ldots, x_n$).
There is a strong connection to G-algebras.

(3.7) Proposition (see [15])
Let A be a $\mathbb{K}$-algebra in the variables $x_1, \ldots, x_n$ subject to the relations in (1). Suppose the ordering condition holds as well. Then the non-degeneracy condition holds if and only if A has a PBW basis (in $x_1, \ldots, x_n$).

Considering the definition of a PBW algebra by Bueso in [5] we see it is similar to that of a G-algebra, except that the non-degeneracy condition is replaced by the requirement of a PBW basis. The proposition above hence proves that these definitions are in fact equivalent.
In other words, each G-algebra has a PBW basis, therefore any $f \in A$ can be written as a linear combination of standard monomials, that is

$$f = \sum_{\alpha} c_\alpha \cdot x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}$$

where $c_\alpha \in \mathbb{K}$ and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0$.
We might also use the multi-index-notation and simply write $f = \sum_\alpha c_\alpha x^\alpha$.

In view of the new approach to G-algebras let us redefine some already introduced definitions.
(3.8) Definition
Let $A = \mathbb{K}\langle x_1, \ldots, x_n \mid x_j x_i = c_{ij} x_i x_j + d_{ij}$ for $1 \leq i \leq j \leq n\rangle$ be a G-algebra as stated in (3.5) and let $\prec$ be a well-ordering on the PBW basis $\{x^\alpha \mid \alpha \in \mathbb{N}_0^n\}$ of A.
Then $\prec$ is called a *monomial ordering* if

$$x^\alpha \prec x^\beta \text{ implies } x^{\alpha+\gamma} \prec x^{\beta+\gamma} \text{ for all } \alpha, \beta, \gamma \in \mathbb{N}_0^n.$$

Since any $f \in A \setminus \{0\}$ can be uniquely written as $f = c_\alpha x^\alpha + \sum_{\beta \neq \alpha} c_\beta x^\beta$ for a given monomial ordering $\prec$ with $c_\alpha, c_\beta \in \mathbb{K}$ and $c_\alpha \neq 0$ such that $f = c_\alpha x^\alpha$ or $x^\beta \prec x^\alpha$ for all $\beta \neq \alpha$ we are able to introduce the ...

$$\begin{aligned} &\ldots \textit{leading monomial} \text{ of } f \quad : \quad lm(f) := x^\alpha \\ &\ldots \textit{leading coefficient} \text{ of } f \quad : \quad lc(f) := c_\alpha. \end{aligned}$$

Furthermore we call $\prec$ an *admissible ordering* if it satisfies the following condition:

$$lm(d_{ij}) \prec x_i x_j \text{ for all } 1 \leq i \lneq j \leq n \text{ with } d_{ij} \neq 0.$$

Note that admissible orderings are well-orderings by definition. To avoid any confusion we mention that whenever we speak of an ordering on a G-algebra, we mean a monomial ordering on the PBW basis, which is also admissible.
As monomial orderings on G-algebras can be seen as restrictions of monomial orders on Mon_n to the set of standard monomials SM_n, it is easy to verify that the two definitions (3.2) and (3.8) coincide.

In order to translate the previous setting to the point of view of integer programming we need some preparation.

(3.9) Definition
Let R be a commutative ring and let A be a (not necessarily commutative) polynomial algebra over R in indeterminates $x_1, \ldots, x_n$. Additionally let $f = \sum_{\alpha \in \mathbb{N}_0^n} c_\alpha x^\alpha \in A \setminus \{0\}$ be a polynomial involving only standard polynomials with $c_\alpha \in R$ and only finitely many $c_\alpha \neq 0$.

1) The set of multi-indices
$$\mathcal{N}(f) := \{\alpha \in \mathbb{N}_0^n \mid c_\alpha \neq 0\}$$
is called the *Newton diagram* of f.

2) The *weighted (total) degree* of f with respect to the weight $\omega = (\omega_1, \ldots, \omega_n)^T \in \mathbb{R}^n$ is defined to be
$$deg_\omega(f) := \max \left\{ \sum_{i=1}^{n} \omega_i \alpha_i \mid \alpha_i \in \mathcal{N}(f) \right\}.$$

For the zero polynomial we set $\mathcal{N}(0) := \emptyset$ and $deg_\omega(0) := -\infty$ for any $\omega \in \mathbb{R}^n$.

(3.10) Example
Consider the algebra
$$A = \mathbb{K}\langle x, y, z \mid yx = xy + y^2, zx = xz + z^3, zy = yz + z^2 \rangle.$$
Then we have $\mathcal{N}(xy + y^2) = \{(1,1,0),\ (0,2,0)\}$, $\mathcal{N}(xz + z^3) = \{(1,0,1),\ (0,0,3)\}$ and $\mathcal{N}(yz + z^2) = \{(0,1,1),\ (0,0,2)\}$.

Given an ordering on an algebra A with relations as in (1) it is possible to construct an admissible ordering such that the non-degeneracy condition is fulfilled. The following result by *Bueso*, *Gómez-Torrecillas* and *Lobillo* [4] gives an algorithm to check whether an algebra is a G-algebra. Moreover it yields a first application of integer programming methods to G-algebras and contains an interesting result for weighted monomial orderings.

(3.11) Proposition
Let

$$A = \mathbb{K}\langle x_1, \dots, x_n \mid x_j x_i = c_{ij} x_i x_j + d_{ij} \text{ for } 1 \leq i \leq j \leq n \rangle$$

with $c_{ij} \in \mathbb{K} \setminus \{0\}$ and d_{ij} as in (3.5) such that the non-degeneracy condition is fulfilled. Further set

$$B := \bigcup_{1 \leq i < j \leq n} \{\alpha - e_i - e_j \mid \alpha \in \mathcal{N}(d_{ij})\}.$$

Then there exists an ordering $\prec$ satisfying the ordering condition (i.e. A is a G-algebra) if and only if the linear programming problem

$$\begin{array}{llcl} \text{Min} & \sum_{j=1}^{n} & \omega_j & \\ \text{s.t.} & \omega_j & \geq & 1,\ j = 1, \dots, n \\ & b^T \cdot \omega & \leq & -1,\ b \in B \end{array} \qquad (2)$$

has a solution. Moreover, given a (not necessarily admissible) monomial well-ordering $\prec$ on A , each solution ω of (2) gives rise to an admissible *weighted degree* ordering $\prec_\omega$ by setting

$$\begin{array}{lll} x^\alpha \prec_\omega x^\beta & \text{if} & deg_\omega(x^\alpha) < deg_\omega(x^\beta) \\ & \text{or} & deg_\omega(x^\alpha) = deg_\omega(x^\beta) \text{ and } x^\alpha \prec x^\beta. \end{array}$$

Note that the proposition implies that for any G-algebra there exists an admissible weighted degree ordering. A more detailed analysis of these orderings will follow in the following sections.

(3.12) Example
Consider again the algebra from the example above, that is

$$A = \mathbb{K}\langle x, y, z \mid yx = xy + y^2, zx = xz + z^3, zy = yz + z^2 \rangle.$$

The ordering condition is fulfilled if and only if $x \succ y$, $x \succ z$ and $y \succ z$. The lexicographical ordering with $x \succ y \succ z$ has this property. Furthermore, applying Proposition (3.11), any solution of the integer programming problem

$$\begin{array}{ll} \text{Min} & \omega_x + \omega_y + \omega_z \\ \text{s.t.} & -\omega_x + \omega_y \leq -1 \\ & -\omega_x + 2\omega_z \leq -1 \\ & -\omega_y + \omega_z \leq -1 \\ & \omega_x, \omega_y, \omega_z \geq 1, \end{array}$$

yields an admissible weighted degree ordering. The vector $\omega = (\omega_x, \omega_y, \omega_z) = (3, 2, 1)$ for example is such a solution, that is, given a monomial well-ordering $\prec$, the induced ordering $\prec_{(3,2,1)}$ is admissible.

— *Elimination of variables* —

We would like to point out another relation between the theory of Gröbner bases and linear programming problems. This is one of the main applications of Gröbner bases, namely the intersection of a given ideal with a given subalgebra, where the subalgebra is generated by a set of indeterminates. This is also known as *elimination of variables.* The key result of this subsection is due to García García, García Miranda and Lobillo in [11], which yields an interface to linear programming.

Consider a G-algebra $A = \langle x_1, \dots, x_n \mid \{x_j x_i = c_{ij} x_i x_j + d_{ij}\}_{1 \le i < j \le n} \rangle$.
In contrast to the commutative case one has to be more considerate with respect to the regarded subalgebra, that is even if the subalgebra is generated by a proper subset of the variables $x_1, \dots, x_n$, it is not automatically a proper subalgebra of A.

(3.13) Example
Consider the G-algebra

$$A = \mathbb{K}\langle x, y, z \mid yx = xy + z, zx = xz + x, zy = yz + y \rangle,$$

subject to the lexicographical order $x \succ y \succ z$. Here the subalgebra generated only by x and y equals A. This is due to the first relation, that is $z = yx - xy \in \mathbb{K}\langle x, y \rangle$.

Hence a more detailed definition of the regarded subalgebra is necessary.

(3.14) Definition
Let $A = \langle x_1, \dots, x_n \mid x_j x_i = c_i x_i x_j + d_{ij}$ for $1 \le i < j \le n\rangle$ be a G-algebra and $I = \{i_1, \dots, i_k\} \subset \{1, \dots, n\}$ an index set with $i_j < i_{j+1}$ for $j = 1, \dots, k-1$. Let A' denote the corresponding subalgebra of A with respect to I, that is A' is generated by $x_{i_1}, \dots, x_{i_k}$.
Then

a) A' is called *admissible* if d_{i_a, i_b} involves only standard monomials in $x_{i_1}, \dots, x_{i_k}$ for $i_1 \le i_a < i_b \le i_k$.

b) If A' is an admissible subalgebra of A, an admissible ordering $\prec$ on A with the property $lm(f) \in A'$ implies $f \in A'$ for all$f \in A \setminus \{0\}$ is called an (admissible) elimination ordering for $\{x_i \mid i \notin I\}$. Moreover, if the subalgebra A'' generated by $\{x_i \mid i \notin I\}$ is also admissible, we call an elimination ordering for $\{x_i \mid i \notin I\}$ an elimination ordering for A''.

As A' inherits its structure from A it is multiplicatively closed and hence a *sub-G-algebra*.

There is an interesting consequence for Gröbner bases ([15]):

(3.15) Lemma
Let A be a G-algebra, A' an admissible subalgebra of A and $I \subseteq A$ a left ideal. If G is a Gröbner basis of I with respect to an elimination ordering $\prec$ for the variables of A not contained in A', then $G \cap A'$ is a Gröbner basis of $I \cap A'$ with respect to $\prec$.

Proof
Let $f \in I \cap A' \subseteq I$. Since G is a Gröbner basis of I, there exists $g \in G$ such that $lm(g)|lm(f)$. Because $lm(f) \in A'$, so is $lm(g)$. By definition of the elimination ordering, $lm(g) \in A'$ implies $g \in A'$. But then it follows from $G \cap A' \subseteq I \cap A'$ that $G \cap A'$ is a Gröbner basis of $I \cap A'$. $\square$

However, while one can always take the lexicographical ordering as an elimination ordering in the commutative case, the existence of elimination orderings in the non-commutative case is not guaranteed as we will see in the next example.

(3.16) Example
For
$$A = \mathbb{K}\langle x, y \mid yx = xy + y^2\rangle,$$
we have $A' = \mathbb{K}[x]$ as an admissible subalgebra. For an admissible ordering $\prec$ on A though, which has so satisfy $y \prec x$, it holds $lm(x + y) = x \in \mathbb{K}[x]$, but $x + y \notin \mathbb{K}[x]$. Hence there is no elimination ordering for y on A.

So, for a given subalgebra A' we wish to decide whether it is admissible and if so how we can show the existence of or even compute an elimination ordering for the variables of A not contained in A'. These questions are answered by the following theorem by García García, García Miranda and Lobillo, see [11].

(3.17) Theorem
Let

$$A = \mathbb{K}\langle x_1, \dots, x_n \mid x_j x_i = c_{ij} x_i x_j + d_{ij} \text{ for } 1 \le i \le j \le n \rangle$$

be a G-algebra. Further, set

$$B := \bigcup_{1 \le i < j \le n} \{e_i + e_j - \alpha \mid \alpha \in \mathcal{N}(d_{ij})\}$$

and let $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$. Then the subalgebra generated by $\{x_j \mid j \in I\}$ is admissible and there exists an elimination ordering for $\{x_i \mid i \notin I\}$ if and only if the linear programming problem

$$\begin{array}{ll} \text{Min} & \sum_{j=1}^{n} \omega_j \\ \text{s.t.} & \omega_j \ge 1,\ j \notin I \\ & \omega_j = 0,\ j \in I \\ & b^T \omega \ge 0,\ b \in B, \end{array}$$

has a solution. Moreover, given an admissible ordering $\prec$ on A , each solution ω of the above linear program gives rise to a weighted degree elimination ordering $\prec_\omega$ by setting

$$\begin{array}{lll} x^\alpha \prec_\omega x^\beta & \text{if} & deg_\omega(x^\alpha) \prec deg_\omega(x^\beta) \\ & \text{or} & deg_\omega(x^\alpha) = deg_\omega(x^\beta) \text{ and } x^\alpha \prec x^\beta. \end{array}$$

Let us take a look at a few examples.

(3.18) Example
Consider Example (3.12) again and say we want to eliminate z. As

$$A = \mathbb{K}\langle x, y, z \mid yx = xy + y^2, zx = xz + z^3, zy = yz + z^2 \rangle,$$

applying Theorem (3.17) leads to the linear programming problem

$$\begin{array}{lrl} \text{Min} & \omega_x + \omega_y + \omega_z & \\ \text{s.t.} & \omega_x & = 0 \\ & \omega_y & = 0 \\ & \omega_z & \ge 1 \\ & -\omega_x + \ \omega_y & \ge 0 \\ & -\omega_x + 2\omega_z & \ge 0 \\ & -\omega_y + \ \omega_z & \ge 0 \\ & \omega_x, \omega_y, \omega_z & \ge 1, \end{array}$$

where the set of constraints is equivalent to the following

$$\begin{aligned} \omega_x &= 0 \\ \omega_y &= 0 \\ \omega_z &\geq 1 \\ 2\omega_z &\geq 0, \end{aligned}$$

which has obviously the optimal solution $(\omega_x, \omega_y, \omega_z) = (0, 0, 1)$ and in particular, we can eliminate z.

(3.19) Example
Consider the G-algebra

$$A = \langle x, y, z \mid yx = xy - z,\ zx = xz + 2x,\ zy = yz - 2y \rangle.$$

Despite which variables we want to eliminate, we will always have to take the following constraints into account

$$\begin{aligned} -\omega_x - \omega_y + \omega_z &\geq 0 \\ -\omega_z &\geq 0. \end{aligned}$$

Hence any subset of the variables containing z and in particular z itself can not be eliminated as this implies to add the constraint $\omega_z \geq 1$ to the ones given above, which leads to infeasibility. Therefore $\omega_z = 0$ (if we want to eliminate other variables) and the remaining candidates for an elimination are x and y but as one can easily see, setting $\omega_x \geq 1$ and/or $\omega_y \geq 1$ contradicts $-\omega_x - \omega_y \geq 0$. Hence there exists no elimination ordering in A.

As stated above, the usual monomial ordering in the commutative case, i.e. the lexicographical ordering, is, in general, not admissible in the non-commutative case, here one typically uses a block-ordering. More precisely, the generating variables are divided into a number different blocks and for each block one chooses a monomial ordering. For instance we can divide the variables $x_1, \ldots, x_n$ into the two blocks $x_1, \ldots, x_r$ and $x_{r+1}, \ldots, x_n$ and fix the block ordering $((x_1, \ldots, x_r)_{dp}, (x_{r+1}, \ldots, x_n)_{lp})$. Here two monomials are compared first by the first block ordering $(x_1, \ldots, x_r)_{dp}$ and in case of a tie one takes the next block ordering into account and so on. Of course, one can also define a block ordering by a sequence of *weighted* monomial orderings.
In order to contribute to this topic in terms of a pattern of grading and balancing of weights for the considered weight vectors so far we will first have to introduce basics of filtrations and gradings as well as Lie algebras and universal enveloping algebras.

§4 Filtrations of G-algebras

This section is based on Levandovskyy [15]. We will present the basics of filtrations and present two different methods to filter a G-algebra. In doing so we will again encounter weight vectors. While we keep this section strictly theoretical we will focus on explicit computations and applications in section 7.

(4.1) Definition
An algebra A is called *filtered* if for every non-negative integer i there is a subspace A_i such that

1) $A_i \subseteq A_j$ if $i \leq j$

2) $A_i \cdot A_j \subseteq A_{i+j}$

3) $A = \bigcup_{i=0}^{\infty} A_i$.

The set $\{A_i \mid i \in \mathbb{N}\}$ is called a *filtration* of A.

(4.2) Definition
An *associated graded algebra* $Gr(A)$ of a filtered algebra A is defined to be

$$Gr(A) = \bigoplus_{i=1}^{\infty} G_i \text{ where } G_i = A_i/A_{i-1} \text{ and } A_{-1} = 0,$$

with the induced multiplication

$$(a_i + A_{i-1}) \cdot (a_j + A_{j-1}) = a_i \cdot a_j + A_{i+j-1}.$$

There are two different kinds of filtrations on A:

— *Weighted degree filtration* —

Let $\prec_\omega$ be a weighted degree ordering on A, i.e. there is an n-tuple of strictly positive weights $\omega = (\omega_1, \ldots, \omega_n)$ and some ordering $\prec$ on A.
Then

$$\alpha \prec_\omega \beta \Leftrightarrow \sum_{i=1}^{n} \omega_i\beta_i < \sum_{i=1}^{n} \omega_i\alpha_i \text{ or, if } \sum_{i=1}^{n} \omega_i\beta_i = \sum_{i=1}^{n} \omega_i\alpha_i, \text{ then } \alpha \prec \beta.$$

Assume that $\omega_1 \geq \ldots \geq \omega_n$ and all the weights are positive integers. Let us define $deg_\omega(x^\alpha) := \omega_1\alpha_1 + \ldots \omega_n\alpha_n$ and call it *weighted degree function* on A. For any polynomial $f \in A$, we define $deg_\omega(f) := deg_\omega(lm(f))$. Note that $deg_\omega(x^\alpha) = 0 \Leftrightarrow \alpha = 0$.

Further note, that in a G-algebra the term $lm(f)$ needs to be treated more carefully, as for the product of two polynomials f and g, it does not hold $lm(fg) = lm(f)lm(g)$ in general. This is due to the fact that a product of two monomials does usually not result in a monomial but in a polynomial. However we have $lm(fg) = lm(lm(f)lm(g))$.
A first property of the weighted degree function is given by the following lemma.

(4.3) Lemma
Let f and g be two polynomials in A.
Then

$$deg_\omega(fg) = deg_\omega(f) + deg_\omega(g).$$

Proof
Let us first consider two monomials x^α, x^β. Then we have

$$deg_\omega(x^\alpha x^\beta) = deg_\omega(x^{\alpha+\beta}) = \sum_{i=1}^{n} \omega_i(\alpha_i + \beta_i) = deg_\omega(x^\alpha) + deg_\omega(x^\beta). \tag{2}$$

Taking two polynomials f and g and keeping the above discussion about the leading monomial in mind we conclude on one hand

$$deg_\omega(fg) = deg_\omega(lm(fg)) = deg_\omega(lm(lm(f)lm(g))).$$

Using the definion of the weighted degree function several times yields on the other hand

$$\begin{aligned} deg_\omega(f) + deg_\omega(g) = deg_\omega(lm(f)) + deg_\omega(lm(g)) &\overset{(2)}{=} deg_\omega(lm(f)lm(g)) \\ &= deg_\omega(lm(lm(f)lm(g))), \end{aligned}$$

where the last equality is again due to the definition on $deg_\omega(\cdot)$. Combining the above yields the assertion. □

Let us construct the first method to filter a G-algebra.
Say A_i is the $\mathbb{K}$-vector space generated by $\{m \in Mon_n(A) \mid deg_\omega(m) \leq i\}$. Hence $A_0 = \mathbb{K}$ and

$$A_{\omega_n} = \begin{cases} \mathbb{K} \oplus \mathbb{K}x_n, & \omega_{n-1} > \omega_n \\ \mathbb{K} \oplus \bigoplus_{m=1}^{n} \mathbb{K}x_m, & \omega_1 = \ldots = \omega_n. \end{cases}$$

Therefore it holds

$$\forall\ 0 \leq i < j \quad A_i \subseteq A_j \subseteq A \text{ and } A = \bigcup_{i=0}^{\infty} A_i.$$

Moreover, Lemma (4.3) implies $A_i \cdot A_j \subseteq A_{i+j}$.

(4.4) Example
Let us pick up the G-algebra given in Example (3.12). Here we have already computed for the G-algebra

$$A = \mathbb{K}\langle x, y, z \mid yx = xy + y^2, zx = xz + z^3, zy = yz + z^2\rangle$$

the weighted degree ordering $\prec_{(3,2,1)}$, where $\prec = \prec_{lp}$.
Now $A_i = \mathbb{K}\langle m \in Mon_n(A) \mid deg_{(3,2,1)}(m) \leq i\rangle$, that is $A_0 = \mathbb{K},\ A_1 = \mathbb{K} \oplus \mathbb{K}z,\ A_2 = A_1 \oplus \mathbb{K}y,\ A_3 = A_2 \oplus \mathbb{K}x \oplus \mathbb{K}zy, \ldots$.

Next consider the sets $G_i = A_i/A_{i-1}$, which consist of all weighted homogeneous elements of weighted degree i in A and $G_0 = A_0 = \mathbb{K}$. By Levandovskyy [15], we have the following.

(4.5) Lemma
Suppose we have an algebra A, where $\forall\ i < j\ deg_\omega(d_{ij}) < deg_\omega(x_i x_j) = \omega_i + \omega_j$. Denote $\bar{x}_i = x_i + A_{i-1}$, then

$$Gr_{deg_\omega}(A) = \bigoplus_{i=1}^{\infty} G_i = \mathbb{K}\langle \bar{x}_1, \ldots, \bar{x}_n \mid \bar{x}_j \bar{x}_i = c_{ij}\bar{x}_i\bar{x}_j \text{ for all } j > i\rangle,$$

that is $Gr_{deg_\omega}(A)$ is isomorphic to a quasi-commutative ring in n indeterminates.

— *Filtration by a monomial ordering* —

Here let $\prec$ be any monomial well-ordering on A.
For $\alpha \in \mathbb{N}^n$ set

$$x^{\prec\alpha} := \{x^\beta \in A \mid x^\beta \prec x^\alpha\}$$

and let A_α be the $\mathbb{K}$-vector space $A_\alpha := span_{\mathbb{K}}(x^{\prec\alpha} \cup \{x^\alpha\})$.
We see that A_α is finitely generated only if there are finitely many x^β with $x^\beta \prec x^\alpha$, i.e. $\prec$ is *finitely supported*. Furthermore it is easy to see that $A_0 = \mathbb{K}$ and

$$\forall\ \beta < \alpha \text{ it holds } A_\beta \subset A_\alpha \subset A \text{ and } A = \bigcup_{\alpha\in\mathbb{N}^n} A_\alpha.$$

Moreover, since $lm(x^\alpha x^\beta) = x^{\alpha+\beta}$ we have $A_\alpha \cdot A_\beta \subseteq A_{\alpha+\beta}$.
Taken together the above, A is a filtered algebra, but as we consider $\mathbb{N}^n$ instead of $\mathbb{N}$, we call it a *multi-filtration*, that is A is a multi-filtered algebra.
In order to display the associated graded algebra we first set

$$\sigma(\alpha) := max_<\{\gamma \mid \gamma < \alpha\}$$

with $\sigma(0) = \emptyset$.
Then for $\alpha \in \mathbb{N}^n$ we have $G_\alpha = A_\alpha / A_{\sigma(\alpha)} = \{x^\alpha\}$ and it follows

$$Gr_{\prec}(A) = \bigoplus_{\alpha \in \mathbb{N}^n} G_\alpha \cong \mathbb{K}\langle \bar{x}_1, \ldots, \bar{x}_n \mid \bar{x}_j \bar{x}_i = c_{ij} \bar{x}_i \bar{x}_j \text{ for all } j > i \rangle,$$

where $\bar{x}_i = x_i + A_{\sigma(e_i)}$ and $e_i \in \mathbb{N}^n$ is the i-th standard basis vector.

While the following section on Lie algebras and the (quantum) universal enveloping algebra of a Lie algebra yields a theoretical application for these filtrations, the upcoming sections afterwards also add a practical approach via the specific computation of weight vectors.

§ 5 Lie Algebras and (Quantum) Universal Enveloping Algebras

As an application of our results we are going to take a look at different kinds of Lie algebras. Particularly interesting for our computations later on will be the the Lie algebra $\mathfrak{sl}_3$ as well as its (quantum) universal enveloping algebra $U(\mathfrak{sl}_3)$ resp. $U_q(\mathfrak{sl}_3)$. Hence we focus on these in the given examples. For a detailed treatment of Lie algebras however we refer the reader to the work by Dixmier [10].

— *Lie Algebras* —

Again let $\mathbb{K}$ be a commutative field.

(5.1) Definition
A *Lie algebra* over a field $\mathbb{K}$ is a $\mathbb{K}$-vector space $\mathfrak{g}$ together with a map

$$[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, (x, y) \mapsto [x, y],$$

termed *Lie bracket*, such that

(1) $[x, y]$ is $\mathbb{K}$-bilinear in x and y

(2) $[x, x] = 0 \ \forall x \in \mathfrak{g}$

(3) $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0 \ \forall x, y, z \in \mathfrak{g}$

(5.2) Remark
Properties (1) and (2) imply that $[y, x] = -[x, y]$ for all $x, y \in \mathfrak{g}$. Property (3) is also known as the Jacoby identity, the idea of which is to get a replacement of the notion of associativity, since in general a Lie algebra is neither commutative nor associative.

For our purposes the *Lie bracket* $[x, y]$ in the definition above will denote

$$[x, y] = xy - yx,$$

which we will also refer to as the *canonical Lie bracket.*
Via the induced mapping, any $\mathbb{K}$-algebra can be given a Lie algebra structure.
Let V be a finite dimensional vector space over a field $\mathbb{K}$, and $\mathrm{End}(V)$ the algebra of endomorphisms of V. We will denote the Lie algebra of $\mathrm{End}(V)$, that is $\mathrm{End}(V)$ endowed with the canonical Lie bracket, by $\mathfrak{gl}(V)$ also known as the general Lie algebra. To see

that $\mathfrak{gl}(V)$ actually is a Lie algebra one has to check the properties (1) – (3). While (1) and (2) are immediate, property (3) requires some calculation which is straightforward and will not be displayed here.
Let $\dim_{\mathbb{K}}(V) = n$. In view of explicit calculations later on, it is more convenient to interpret $\mathfrak{gl}(V)$ as the set of all square matrices of order n with elements in $\mathbb{K}$. Hence we have the canonical base of $\mathfrak{gl}(n, \mathbb{K}) = \mathfrak{gl}(V)$, namely $\{E_{ij} \mid 1 \leq i, j \leq n\}$ where E_{ij} denotes the matrix with all entries equal to zero except for the entry at position (i, j), which equals 1. Furthermore we can give an explicit expression of the Lie bracket

$$[E_{ij}, E_{kl}] = \delta_{jk} E_{il} - \delta_{il} E_{kj}$$

where δ_{ij} is the Kronecker delta.

(5.3) Definition
Let g be a Lie algebra as above. A subspace $\mathfrak{h} \subseteq \mathfrak{g}$ is called a *Lie subalgebra* of $\mathfrak{g}$, if it is closed with respect to the Lie bracket $[\cdot, \cdot]$ on $\mathfrak{g}$. That is, for any $h_1, h_2 \in \mathfrak{h}$ it holds $[h_1, h_2] \in \mathfrak{h}$.

Moreover we call a Lie subalgebra $\mathfrak{i}$ of $\mathfrak{g}$ an *ideal* of $\mathfrak{g}$ if

$$[x, y] \in \mathfrak{i} \text{ for all } x \in \mathfrak{g},\ y \in \mathfrak{i}.$$

For instance the set of the $x \in \mathfrak{gl}(V)$ whose trace $tr(x)$ is zero is an ideal of $\mathfrak{gl}(V)$ and denoted by $\mathfrak{sl}(V)$. In particular, $\mathfrak{sl}(V)$ itself is a Lie algebra.
Using the notation $\mathfrak{gl}(n, \mathbb{K})$ as we did above, we refer to the corresponding Lie algebra of $\mathbf{M}_n(\mathbb{K})$, the algebra of square matrices of order n with elements in $\mathbb{K}$. The notation $\mathfrak{sl}(n, \mathbb{K})$ is self-explanatory.

Let us take a look at important examples.

(5.4) Examples

1. $\mathfrak{sl}_2$
   The Lie algebra $\mathfrak{sl}(2, \mathbb{K})$ will be denoted by $\mathfrak{sl}_2$. The elements

$$e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = E_{12}, \quad f = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = E_{21}, \quad h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = E_{11} - E_{22}$$

   form a basis of $\mathfrak{sl}_2$. We have

$$[h, e] = 2e, \quad [h, f] = -2f, \quad [e, f] = h.$$

2. $\mathfrak{sl}_3$
   One can develop a basis of the vector space $\mathfrak{sl}_3$ from the one of $\mathfrak{sl}_2$, namely

$$\begin{aligned} e_{1,2} = E_{12}, \quad e_{1,3} &= E_{13}, \quad e_{2,3} = E_{23}, \\ f_{1,2} = E_{21}, \quad f_{1,3} &= E_{31}, \quad f_{2,3} = E_{32} \end{aligned}$$

   and

$$h_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad h_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

A general structure of the basis of $\mathfrak{sl}_{n+1}$ is given by

$$\begin{aligned} e_{i,j} &= E_{ij}, \ f_{i,j} = E_{ji} \text{ for } 1 \le i < j \le n+1 \\ \text{and} \quad h_i &= E_{ii} - E_{i+1i+1} \text{ for } 1 \le i \le n, \end{aligned} \tag{3}$$

see [19] for example.

As mentioned before, the general Lie algebra lacks associativity. Hence many known results of associative algebras are not directly applicable to Lie algebras in this setting. This problem leads us to the *universal enveloping algebra of a Lie algebra.*

— *Universal Enveloping Algebras* —

An explicit construction and analysis of the universal enveloping algebra of a Lie algebra along with a proof of the famous *Poincaré-Birkhoff-Witt-Theorem* (shortly PBW-Theorem), which we will also make use of, can be found for example in *Introduction to Lie Algebras and Representation Theory*, [13].
Furthermore, along with an abstract view of universal enveloping algebras, we will also focus on a constructive way as this is more convenient for calculations.

Consider two Lie algebras, $\mathfrak{g}_1$ and $\mathfrak{g}_2$. Then a *morphism* of Lie algebras is a $\mathbb{K}$-linear map $\varphi : \mathfrak{g}_1 \to \mathfrak{g}_2$, which is compatible with the Lie-brackets of $\mathfrak{g}_1$ and $\mathfrak{g}_2$, that is

$$\varphi([x, y]) = [\varphi(x), \varphi(y)]$$

for all $x, y \in \mathfrak{g}_1$.

In view of the following definition recall, that any $\mathbb{K}$-algebra can be viewed as a Lie algebra via the canonical Lie-bracket.

(5.5) Definition
Let U be an associative $\mathbb{K}$-algebra and $\mathfrak{g}$ a Lie algebra. One calls U a universal enveloping algebra for $\mathfrak{g}$ if it is endowed with a morphism of Lie algebras $\varphi : \mathfrak{g} \to U$ (w.r.t. the canonical Lie-bracket on U) satisfying the following universal property:
For any $\mathbb{K}$-algebra V and any morphism of Lie algebras $\psi : \mathfrak{g} \to V$ (w.r.t. the canonical Lie-bracket on V) there exists a unique $\mathbb{K}$-algebra morphism $\theta : U \to V$ such that the diagram

$$\begin{tikzcd}
\mathfrak{g} \arrow[r, "\varphi"] \arrow[dr, "\psi"'] & U \arrow[d, dashed, "\theta"] \\
& V
\end{tikzcd}$$

commutes.

Obviously U is unique up to unique isomorphism: replace V in the diagram above by any associative $\mathbb{K}$-algebra U' with the same property as U. Then there exists besides $\theta : U \to U'$ also a unique $\mathbb{K}$-algebra morphism $\theta' : U' \to U$ and hence we can construct an isomorphism $\phi : U \to U'$ via $\phi = \theta' \circ \theta$. Hence we will denote the universal enveloping algebra for a Lie algebra $\mathfrak{g}$ by $U(\mathfrak{g})$.
To see that $U(\mathfrak{g})$ actually always exists, one can construct it in the following way, for details see Dixmier [10] for example. Let a $\mathbb{K}$-basis of $\mathfrak{g}$ be given by $\{x_j \mid j \in J\}$, where J is an index set. Then by setting $U(\mathfrak{g})$ as the quotient of the free $\mathbb{K}$-algebra $\mathbb{K}\langle x_j \mid x_j \in J\rangle$ by the ideal I generated by the elements of the form

$$x_i x_j - x_j x_i - [x_i, x_j]$$

we obtain an associative $\mathbb{K}$-algebra satisfying the requirements of a universal enveloping algebra for $\mathfrak{g}$.
Surprisingly, we even have the following result.

(5.6) Theorem (PBW-Theorem, [10])
Let $(x_1, \ldots, x_n)$ be any ordered basis of a finite dimensional Lie algebra $\mathfrak{g}$ over a field $\mathbb{K}$. Then

$$\{x_1^{l_1} \cdots x_n^{l_n} \mid l_1, \ldots, l_n \in \mathbb{N}_0\}$$

is a basis for $U(\mathfrak{g})$.

For a Lie algebra $\mathfrak{g}$ over $\mathbb{K}$ we see immediately on one side that $U(\mathfrak{g})$ is a domain and on the other side, that the mapping $\varphi : \mathfrak{g} \to U(\mathfrak{g})$ with respect to the canonical Lie-bracket on $U(\mathfrak{g})$ is injective, hence $U(\mathfrak{g})$ envelops $\mathfrak{g}$.

(5.7) Examples
Let us take on the examples from above. The PBW-Theorem tells us, that a basis of

1. $U(\mathfrak{sl}_2)$ is given by $\{f^m h^l e^n \mid n, m, l \in \mathbb{N}_0\}$ and
2. $U(\mathfrak{sl}_3)$ is given by $\{f_{1,2}^{m_1} f_{1,3}^{m_2} f_{2,3}^{m_3} h_1^{l_1} h_2^{l_2} e_{1,2}^{n_1} e_{1,3}^{n_2} e_{2,3}^{n_3} \mid m_i, l_i, n_i \in \mathbb{N}_0\}$.

This result is of great importance for explicit calculations in section 7, where we will use these bases with the corresponding relations to construct explicit admissible weight vectors allowing us to view these algebras as G-algebras. Doing so we mainly focus on the algebras in the following subsection which present what we have considered so far in a more general context.

— *Quantum Universal Enveloping Algebras* —

In what follows we deal with a more general treatment of the universal enveloping algebra of a Lie algebra, the so called *quantum* universal enveloping algebra of a Lie algebra over $\mathbb{K} = \mathbb{C}$. Simply speaking we additionally take a fixed parameter $q \in \mathbb{C}$ into account that takes part in the relations of a non-commutative algebra A.

We will start with a typical example, not quite in the context of Lie algebras.
For $0 \neq q \in \mathbb{C}$ the quantum plane $\mathcal{O}_q(\mathbb{C}^2)$ is defined by

$$\mathcal{O}_q(\mathbb{C}^2) := \mathbb{C}\langle x, y\rangle / I_q,$$

where I_q is the two-sided ideal generated by $yx - qxy$. Here we have another simple example of a G-algebra.
This concept can of course be generalized to $\mathcal{O}_q(\mathbb{C}^n) = \mathbb{C}\langle x_1, \dots, x_n\rangle / I_{\mathbf{q}}$, where $\mathbf{q}$ contains the parameters q_{ij} subject to the relations $x_j x_i = q_{ij} x_i x_j$ and $I_{\mathbf{q}}$ is the corresponding ideal generated by the elements $x_j x_i - q_{ij} x_i x_j$.

In the context of Lie algebras we will start with $U_q(\mathfrak{sl}_2)$. Again we consider the canonical Lie-bracket.

(5.8) Definition
Let $0 \neq q \in \mathbb{C}$ such that $q^4 \neq 1$. We define $U_q(\mathfrak{sl}_2)$ to be the quotient of the free

$\mathbb{C}$-algebra $\mathbb{C}\langle e, f, k, k^{-1}\rangle$ and the two-sided ideal corresponding to the relations

$$\begin{aligned} kek^{-1} &= q^2 e \\ kfk^{-1} &= q^{-2} f \\ [e,f] &= \frac{k^2 - k^{-2}}{q^2 - q^{-2}} \\ kk^{-1} &= k^{-1}k = 1. \end{aligned}$$

Note that this is not a generalization of $U(\mathfrak{sl}_2)$ meaning that replacing q with specific value does not lead us back to $U(\mathfrak{sl}_2)$.
We will continue with a detailed analysis of $U_q(\mathfrak{sl}_3)$. For this let us first review the universal enveloping algebra of $\mathfrak{sl}_3$.

(5.9) Example ($U(\mathfrak{sl}_3)$)
Instead of the representation of this algebra given in Example (5.7), we will draw on the one given in `Singular::Plural (ncalg.lib)` [12]
The universal enveloping algebra of the Lie algebra $\mathfrak{sl}_3$ of 3×3 traceless matrices is generated by

$$x_\alpha, x_\beta, x_\gamma, y_\alpha, y_\beta, y_\gamma, h_\alpha, h_\beta$$

subject to the relations

$$\begin{array}{lll}
[x_\alpha, x_\beta] = x_\gamma, & [x_\alpha, y_\alpha] = h_\alpha, & [x\alpha, y_\gamma] = -y_\beta, \\
[x_\alpha, h_\alpha] = -2x_\alpha, & [x_\alpha, h_\beta] = x_\alpha, & [x_\beta, y_\beta] = h_\beta, \\
[x_\beta, y_\gamma] = y_\alpha, & [x_\beta, h_\alpha] = x_\beta, & [x_\beta, h_\beta] = -2x_\beta, \\
[x_\gamma, y_\alpha] = -x_\beta, & [x_\gamma, y_\gamma] = h_\alpha + h_\beta, & [x_\gamma, y_\beta] = x_\alpha, \\
[x_\gamma, h_\alpha] = -x_\gamma, & [x_\gamma, h_\beta] = -x_\gamma, & [y_\alpha, y_\beta] = x_\alpha, \\
[y_\alpha, h_\alpha] = 2y_\alpha, & [y_\alpha, h_\beta] = -y_\alpha, & [y_\beta, h_\alpha] = -y_\beta, \\
[y_\beta, h_\beta] = 2y_\beta, & [y_\gamma, h_\alpha] = y_\gamma, & [y_\gamma, h_\beta] = y_\gamma,
\end{array}$$

where $[x, y] := xy - yx$.
For those familiar with the notion of Cartan matrices, the relations above correspond to the Cartan matrix

$$A_2 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

As a fact this correspondence is unique, hence we will as well use the notation $U_q(A_2)$.
What we did not mention so far is that our treatment of a universal enveloping algebra of a Lie algebra is a spacial case of the one given by Serre resp. the famous Chevalley-Serre relations. A detailed and more general attempt can for instance be found in [19]

by Yamane. However the knowledge of Cartan matrices in this context only affects the explicit construction of some examples and, as before, not the underlying theory of this work or the results of our upcoming computations.
In fact, besides A_2, we will only take the Cartan matrix

$$B_2 = \begin{pmatrix} 2 & -2 \\ -1 & 2 \end{pmatrix}$$

into account.

To be more specific, we will at least present how to obtain the given relations from a Cartan matrix of type $A_n = (a_{ij})_{i,j=1}^n$ in general. In this work we only consider the case $n = 2$. First set

$$a_{ij} = \begin{cases} 2, & \text{if } i = j, \\ -1, & \text{if } |i-j| = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Similar to the structure of $\mathfrak{sl}_\mathfrak{n}$ given in (3) one can prove for $U(\mathfrak{sl}_\mathfrak{n})$, see [19],

$$U(\mathfrak{sl}_\mathfrak{n}) = \mathbb{K}\langle x_i, h_i, y_i \mid 1 \leq i \lneq n \rangle / I,$$

where the two-sided ideal I is represented by the relations

$$\begin{aligned} h_i h_j - h_j h_i &= x_i y_j - y_j x_i = 0 \\ x_i y_i - y_i x_i &= h_i \\ h_i x_j - x_j h_i &= a_{ji} x_j \\ h_j y_i - y_i h_j &= -a_{ji} y_j \end{aligned}$$

and

$$(ad(x_i))^{1-a_{ji}}(x_j) = 0 \text{ resp. } (ad(y_i))^{1-a_{ji}}(y_j) = 0 \tag{4}$$

for $i \neq j$, where $ad(\cdot)(\cdot)$ is defined via

$$ad(x)(y) = xy - yx.$$

Hence the equations (4) can be thought of an interlacing of the canonical Lie-bracket and can be written as

$$\begin{aligned} \sum_{k=1}^{1-a_{ij}} (-1)^k \binom{1-a_{ij}}{k} x_i^{1-a_{ij}-k} x_j x_i^k &= 0 \\ \sum_{k=1}^{1-a_{ij}} (-1)^k \binom{1-a_{ij}}{k} y_i^{1-a_{ij}-k} y_j y_i^k &= 0. \end{aligned}$$

(5.10) Remark

One has to be careful with the given expression of the generating variables of $U(\mathfrak{sl}_3)$. For instance in the example above one can also write

$$\begin{aligned} &x_\alpha = e_{1,2}, && x_\beta = e_{2,3}, && x_\gamma = e_{1,3},\\ &y_\alpha = f_{1,2}, && y_\beta = f_{2,3}, && y_\gamma = f_{1,3},\\ &h_\alpha = h_1, && h_\beta = h_2. \end{aligned}$$

where $e_{i,j}, f_{i,j}$ and h_i are given in (5.4). However we want to draw a clear line in view of the following examples to avoid confusion. There we will also encounter generators denoted by e_{ij} for example, but we wish to stress that $e_{ij} \neq e_{i,j}$.

Now we will take this a few steps further by considering the *quantum universal enveloping algebra* $U_q(\mathfrak{sl}_3)$. Following [17] Example 3.4 where the general case $U_q(\mathfrak{sl}_{\mathfrak{n}+1})$ is treated, we focus on $\mathfrak{sl}_3$ as the underlying Lie algebra. See also [19] for details, especially for the construction of the generating variables below in context of a Cartan matrix. Again we focus on relations corresponding to A_2 at first.

Now, let q be a complex number, such that $q^8 \neq 1$. Instead of examining $U_q(\mathfrak{sl}_3)$ directly, that is taking the $\mathbb{C}$-basis

$$f_{12}^{n_1}, f_{13}^{n_2}, f_{23}^{n_3}, k_1^{\ell_1}, k_2^{\ell_2}, e_{12}^{m_1}, e_{13}^{m_2}, e_{23}^{m_3}$$

with $n_i, m_i \in \mathbb{N}$ and $\ell_i \in \mathbb{Z}$, it is sometimes more convenient or even necessary to avoid negative exponents and consider $V_q(\mathfrak{sl}_3)$. The algebra $V_q(\mathfrak{sl}_3)$ results from $U_q(\mathfrak{sl}_3)$ by setting $l_i := k_i^{-1}$ and regard the generators

$$f_{12}, f_{13}, f_{23}, k_1, k_2, l_1, l_2, e_{12}, e_{13}, e_{23}$$

subject to the relations:

$$\begin{aligned} e_{13}e_{12} &= q^{-2}e_{12}e_{13}, & f_{13}f_{12} &= q^{-2}f_{12}f_{13},\\ e_{23}e_{12} &= q^{2}e_{12}e_{23} - qe_{13}, & f_{23}f_{12} &= q^{2}f_{12}f_{23} - qf_{13},\\ e_{23}e_{13} &= q^{-2}e_{13}e_{23}, & f_{23}f_{13} &= q^{-2}f_{13}f_{23},\\ e_{12}f_{12} &= f_{12}e_{12} + \frac{(k_1^2 - l_1^2)}{(q^2 - q^{-2})}, & e_{12}k_1 &= q^{-2}k_1e_{12},\\ k_1f_{12} &= q^{-2}f_{12}k_1, & e_{12}k_2 &= qk_2e_{12},\\ k_2f_{12} &= qf_{12}k_2, & e_{12}f_{13} &= e_{12}f_{13} + qf_{23}k_1^2,\\ e_{13}k_1 &= q^{-1}k_1e_{13}, & k_1f_{13} &= q^{-2}f_{13}k_1,\\ e_{12}f_{23} &= e_{12}f_{23}, & e_{13}k_2 &= q^{-2}k_2e_{13},\\ k_2f_{13} &= q^{-1}f_{13}k_2, & e_{13}f_{12} &= e_{13}f_{12} - q^{-1}l_1^2e_{23}, \end{aligned}$$

$$\begin{aligned}
&e_{23}k_1 = qk_1e_{23}, && k_1f_{23} = qf_{23}k_1,\\
&e_{13}f_{13} = e_{13}f_{13} - \frac{k_1^2k_2^2 - l_1^2l_2^2}{q^2-q^{-2}}, && e_{23}k_2 = q^{-2}k_2e_{23},\\
&k_2f_{23} = q^{-2}f_{23}k_2, && e_{12}l_1 = q^2l_1e_{12},\\
&l_1f_{12} = q^2f_{12}l_1, && e_{13}f_{23} = e_{13}f_{23} + qk_2^2e_{12},\\
&e_{12}l_2 = q^{-1}l_2e_{12}, && l_2f_{12} = q^{-1}f_{12}l_2,\\
&e_{23}f_{12} = e_{23}f_{12}, && e_{13}l_1 = ql_1e_{13},\\
&l_1f_{13} = qf_{13}l_1, && e_{23}f_{13} = e_{23}f_{13} - q^{-1}f_{12}l_2^2,\\
&e_{13}l_2 = ql_2e_{13}, && l_2f_{13} = qf_{13}l_2,\\
&e_{23}l_1 = q^{-1}l_1e_{23}, && l_1f_{23} = q^{-1}f_{23}l_1,\\
&e_{23}f_{23} = e_{23}f_{23} + \frac{k_2^2 - l_2^2}{q^2-q^{-2}}, && e_{23}l_2 = q^2l_2e_{23},\\
&l_2f_{23} = q^2f_{23}l_2, && l_1k_1 = l_1k_1,\\
&l_2k_1 = l_2k_1, && k_2k_1 = k_2k_1,\\
&l_1k_2 = l_1k_2, \qquad l_2k_2 = l_2k_2, && l_2l_1 = l_2l_1.
\end{aligned}$$

Following [5] Chapter 7, one may verify that a $\mathbb{C}$-basis of $V_q(\mathfrak{sl}_3)$ is given by

$$f_{12}^{n_1}, f_{13}^{n_2}, f_{23}^{n_3}, k_1^{a_1}, k_2^{a_2}, l_1^{b_1}, l_2^{b_2}, e_{12}^{m_1}, e_{13}^{m_2}, e_{23}^{m_3}$$

where $m_i, n_i, a_j, b_j \in \mathbb{N}$.
By Yamane [19] we can write $U_q(\mathfrak{sl}_3)$ as homomorphic image of $V_q(\mathfrak{sl}_3)$:
Setting $I = \langle k_1l_1 - 1, k_2l_2 - 1\rangle$ as a left ideal, we obtain

$$U_q(\mathfrak{sl}_3) \cong V_q(\mathfrak{sl}_3)/I.$$

Using a different Cartan matrix yields other corresponding relations of course, however the results maintain the same.

The long list of relations above is not only displayed to show the complexity of the underlying structure but is actually very useful in view of Theorem (3.11) for the following sections. Among other we will discuss the computation of weight vectors that will lead to $U_q(A_2), V_q(A_2), U_q(B_2)$ or $V_q(B_2)$ being graded in the upcoming section. In section 7 we additionally focus on weight vectors for filtrations, admissible weighted monomial orderings and elimination orderings.

§ 6 The Pattern of Grading

Consider a non-commutative algebra

$$A := \mathbb{K}\langle x_1, \dots, x_n \mid [x_j, x_i] = d_{ij} \text{ for } 1 \leq i < j \leq n\rangle,$$

where $\mathbb{K}$ is a field and d_{ij} is an element of $\operatorname{span}_{\mathbb{K}}\{x_1^{\alpha_1} \cdots x_n^{\alpha_n} \mid \alpha_i \in \mathbb{N}_0, 1 \leq i \leq n\}$.
Usually we deal with relations of the form $yx = cxy + d(x, y)$ where $c \in \mathbb{K}$, but since we are focusing on weighted orderings in this section and $deg(c) = 0$ for any $c \in \mathbb{K} \setminus \{0\}$ with respect to any monomial ordering, we will set each $c = 1$ for ease of notation. Hence the term $[x_j, x_i] = d_{ij}$ in the definition of A is sufficient for our purposes.
Furthermore the d_{ij} take the form

$$d_{ij} = \sum_l d_{ijl}$$

where $d_{ijl} \in Mon_n$ with $lc(d_{ijl}) = 1$ for each l, since we are interested in the support of the polynomials d_{ijl}, that is in terms with coefficient $\neq 0$. As before we have to make some assumptions on the weighted orderings.

(6.1) Definition
A weighted monomial ordering $\prec_\omega$ for a weight vector ω is called *admissible* if

$$x_i x_j \succeq_\omega d_{ijl}$$

for each $1 \leq i < j \leq n$ and each l. More precisely

$$\omega_i + \omega_j \geq \max_l\{deg_\omega(d_{ijl})\}$$

has to hold for each $1 \leq i < j \leq n$ and each l.
We will also refer to such an ω as an *admissible weight vector* and call the set of all admissible weight vectors the *feasible region.*

We are particularly interested in a specific subset of the admissible weighted monomial orderings, namely those that will lead to A being a *graded* algebra. In the notation of the previous definition this means that we are looking for admissible weight vectors ω that satisfy the equations

$$\omega_i + \omega_j = deg_\omega(d_{ijl})$$

for each $1 \leq i < j \leq n$ and each l.

Let us take up again some of the previous examples.

(6.2) Examples

a) *Weyl algebra*

For $A = \mathbb{K}\langle x, \partial \mid \partial x = x\partial + 1\rangle$ any admissible weight vector $\omega = (\omega_x, \omega_\partial)$ has to satisfy

$$\omega_x + \omega_\partial \geq 0.$$

Hence the feasible region is given by

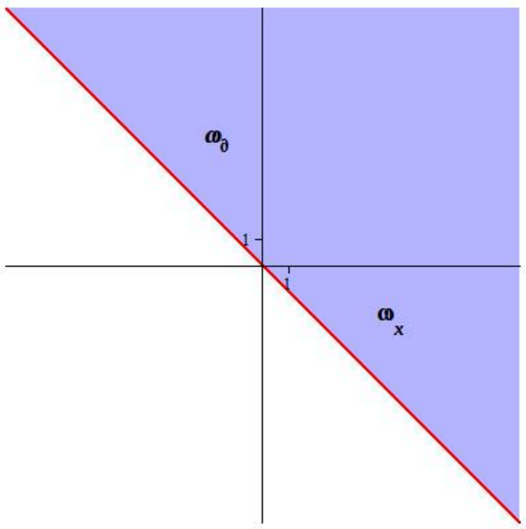

Figure 2: Grading of the Weyl algebra

The red line indicates the weight vector $\omega = (\omega_x, -\omega_x)$, that is for $\omega_x = -\omega_\partial$ the Weyl algebra is graded.

b) *Shift algebra*

For $A = \mathbb{K}\langle k, s \mid sk = ks + s\rangle$ any admissible weight vector $\omega = (\omega_k, \omega_s)$ has to satisfy

$$\omega_k + \omega_s \geq \omega_s \quad \Leftrightarrow \quad \omega_k \geq 0,\ \omega_s \text{ arbitrary}$$

and here the feasible region is given by

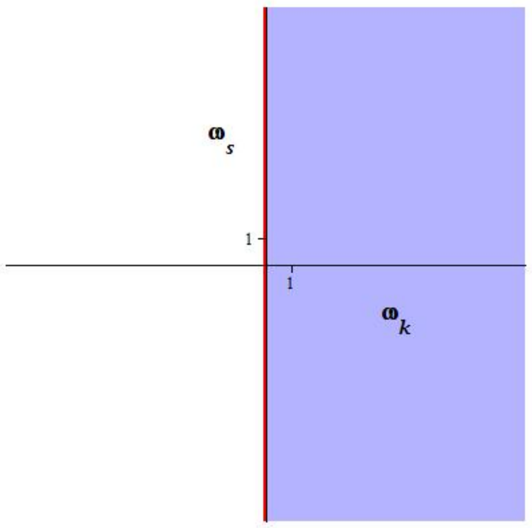

Figure 3: Grading of the Shift algebra

The red line indicates the weight vector $\omega = (0, \omega_s)$, that is for $\omega_k = 0$ the Shift algebra is graded.

c) In both examples above the weight vectors that led to a graded algebra were given by the border of the feasible region. This is not true in general though. Consider the algebra
$$A = \mathbb{K}\langle x, y \mid yx = xy + x + y\rangle.$$
Here an admissible weight vector $\omega = (\omega_x, \omega_y)$ has to satisfy

$$\omega_x + \omega_y \geq \omega_x \quad \text{and} \quad \omega_x + \omega_y \geq \omega_y,$$

and hence the feasible region is given by the intersection of the two planes

$$\{(\omega_x, \omega_y) \mid \omega_x \in \mathbb{R}, \omega_y \geq 0 \ \text{ and } \{(\omega_x, \omega_y) \mid \omega_x \geq 0, \omega_y \in \mathbb{R}\},$$

that is the positive quadrant:

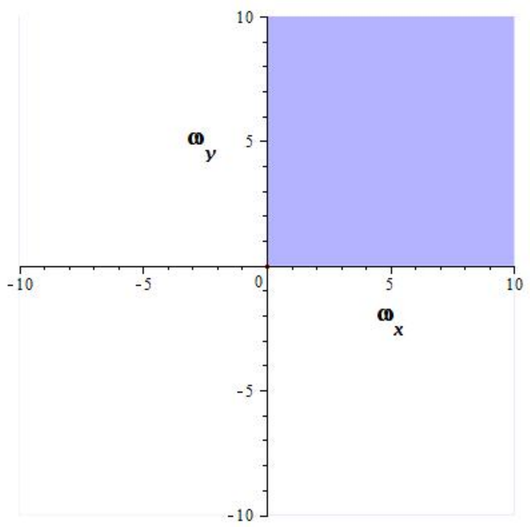

Figure 4: Grading of A

However it is easy to see, that the only possible grading of A is given by the trivial one, that is $\omega = (0, 0)$.

The main issue of this section is to determine a pattern of those weight vectors that imply a grading of the given algebra A in the set of all weight vectors in the feasible region. One might deduce the following from the given examples:

(6.3) Observation

Consider any admissible weight vector ω for a given algebra A. Then each relation $[x_j, x_i] = d_{ij}$ in the definition of A gives us a set of constraints with respect to ω, i.e.

$$\omega_i + \omega_j \geq \max_l \{deg_\omega(d_{ijl})\}.$$

As ω leads to A being a graded algebra if and only if

$$\omega_i + \omega_j = deg_\omega(d_{ijl})$$

for each $1 \leq i < j \leq n$ and each l, we obtain a system of linear equations in the variables $\omega_1, \ldots, \omega_n$. Solving this system leads us to those weight vectors in question.
Geometrically speaking, these weight vectors are given by the intersection of *all* equations bounding the feasible region. This intersection can either result in a single point, a line, a plane ... etc., depending on the number of generators of A and the corresponding relations.
Furthermore, note that the trivial grading, i.e. $\omega = 0$, is always possible. Hence each border of the feasible region has to cross the origin and whenever the intersection of all borders result in a single point, this has to be the origin.

In the examples of (6.2) we have seen all possible outcomes for the two-dimensional case (where A is non-commutative), i.e. there is only one restriction, yielding a line, or there are two restrictions yielding one line each which intersect at the origin.
Let us continue with some examples of higher dimension:

(6.4) Examples

d) $U(\mathfrak{sl}_2)$

Here $A = \mathbb{K}\langle e, f, h \mid [e, f] = h, [e, h] = -2e, [f, h] = 2f\rangle$ and for $\omega = (\omega_e, \omega_f, \omega_h)$ we have to consider the inequalities

$$\begin{aligned} \omega_e + \omega_f &\geq \omega_h \\ \omega_e + \omega_h &\geq \omega_e \\ \omega_f + \omega_h &\geq \omega_f. \end{aligned}$$

which are equivalent to $\omega_e + \omega_f \geq \omega_h$ and $\omega_h \geq 0$. Hence the feasible region is given by

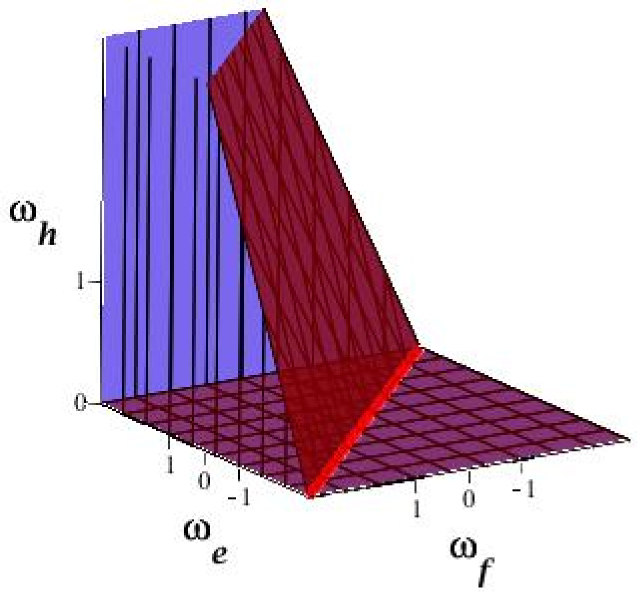

Figure 5: Grading of $U(\mathfrak{sl}_2)$

The red line, i.e. the intersection of the two red planes which represent the borders of the two inequalities, indicates the weight vector $(\omega_e, -\omega_e, 0)$, that is for $\omega_f = -\omega_e$ and $\omega_h = 0$ the algebra $U(\mathfrak{sl}_2)$ is graded.

e) $U(\mathfrak{sl}_3)$
Using the the relations presented in Example (5.9) we conclude similar to the previous example

$$\begin{aligned}\omega &= (\omega_{x_\alpha}, \omega_{x_\beta}, \omega_{x_\gamma}, \omega_{y_\alpha}, \omega_{y_\beta}, \omega_{y_\gamma}, \omega_{h_\alpha}, \omega_{h_\beta}) \\ &= (t,\ -2t,\ -t,\ -t,\ 2t,\ t,\ 0,\ 0),\end{aligned}$$

where $t \in \mathbb{R}$, that is the set of all admissible weight vectors, that will lead to $U(\mathfrak{sl}_3)$ being graded are (again) on one straight line. Furthermore note, that here we also have $\omega_{x_j} = -\omega_{y_j}$.

f) Let $A = \mathbb{K}\langle\ x, y, z \mid yx = qxy + z, zx = qxz + z^3, yz = qzy + y^2\rangle$, where $q \in \mathbb{K}^*$. Here the system of inequalities to be considered is given by

$$\begin{aligned}\omega_x + \omega_y &\geq \omega_z \\ \omega_x + \omega_z &\geq 3\omega_x \Leftrightarrow \omega_z \geq 2\omega_x \\ \omega_z + \omega_y &\geq 2\omega_y \Leftrightarrow \omega_z \geq \omega_y\end{aligned}$$

Replacing each inequality by an *equality* yields a linear system of three irredundant equations in three indeterminates, hence the origin $\omega = (0, 0, 0)$ is the single solution and therefore the only possible grading of A is the trivial one.

g) *The Quantum coordinate* $\mathbb{K}$*-algebra of* 2×2 *matrices* $\mathcal{O}_q(M_2(\mathbb{K}))$
Let $A = \mathcal{O}_q(M_2(\mathbb{K})) := \mathbb{K}\langle a, b, c, d\rangle / I_{\mathbf{q}}$, where $I_{\mathbf{q}}$ is generated by the elements corresponding to the relations:

$$\begin{aligned}ba = qab \quad &, \quad ca = qac \\ cb = qbc \quad &, \quad da = ad + (q + q^{-1})bc \\ db = qbd \quad &, \quad dc = qcd\end{aligned}$$

with $0 \neq q \in \mathbb{K}$. Note that the structure corresponds to the one given at the beginning of subsection 5 of the previous section. Now there is only one inequality defining the feasible region, i.e. $\omega_a + \omega_d \geq \omega_b + \omega_c$. Hence the feasible region is 4-dimensional, which gives us a 3-dimensional object containing the weight vectors $\omega = (\omega_a, \omega_b, \omega_c, \omega_d)$ which will lead to A being graded, namely those that satisfy $\omega_d = \omega_b + \omega_c - \omega_a$.

Additionally let us examine some more complicated examples, namely the quantum universal algebras described in previous section.

(6.5) Examples

h) $\mathbf{V_q(A_2)}$ vs. $\mathbf{U_q(A_2)}$

Consider the relations of section 5 on page 41 and set the weight vector ω to

$$\omega = (\omega_{f_{12}}, \omega_{f_{13}}, \omega_{f_{23}}, \omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_{12}}, \omega_{e_{13}}, \omega_{e_{23}}).$$

We go straight to the resulting system of equalities

$$\begin{aligned}
&\omega_{e_{12}} + \omega_{e_{23}} = \omega_{e_{13}}, && \omega_{f_{12}} + \omega_{f_{23}} = \omega_{f_{13}},\\
&\omega_{e_{12}} + \omega_{f_{12}} = 2\omega_{k_1}, && \omega_{e_{23}} + \omega_{f_{23}} = 2\omega_{k_2},\\
&\omega_{e_{12}} + \omega_{f_{12}} = 2\omega_{l_1}, && \omega_{e_{23}} + \omega_{f_{23}} = 2\omega_{l_2},\\
&\omega_{e_{23}} + \omega_{f_{13}} = \omega_{f_{12}} + 2\omega_{l_2}, && \omega_{e_{12}} + \omega_{f_{13}} = 2\omega_{k_1} + \omega_{f_{23}},\\
&\omega_{e_{13}} + \omega_{f_{13}} = 2\omega_{k_1} + 2\omega_{k_2}, && \omega_{e_{13}} + \omega_{f_{23}} = 2\omega_{k_2} + \omega_{e_{12}},\\
&\omega_{e_{13}} + \omega_{f_{12}} = 2\omega_{l_1} + \omega_{e_{23}}, && \omega_{e_{13}} + \omega_{f_{13}} = 2\omega_{l_1} + 2\omega_{l_2},
\end{aligned}$$

the solution of which is

$$\begin{aligned}
\omega_{VA_2} &= (\omega_{f_{12}}, \omega_{f_{13}}, \omega_{f_{23}}, \omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_{12}}, \omega_{e_{13}}, \omega_{e_{23}})\\
&= (a-b,\ a,\ b,\ c,\ d,\ c,\ d,\ b-a+2c,\ 2c+2d-a,\ 2d-b),
\end{aligned}$$

where $a, b, c, d \in \mathbb{R}$.

Solving the similar system of equations corresponding to $U_q(A_2)$ however, that is adding the equalities

$$\omega_{l_i} = -\omega_{k_i} \text{ for } i = 1, 2,$$

to the ones above, yields the weight vector

$$\begin{aligned}
\omega_{UA_2} &= (\omega_{f_{12}}, \omega_{f_{13}}, \omega_{f_{23}}, \omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_{12}}, \omega_{e_{13}}, \omega_{e_{23}})\\
&= (a-b,\ a,\ b,\ 0,\ 0,\ 0,\ 0,\ b-a,\ -a,\ -b),
\end{aligned}$$

where $a, b \in \mathbb{R}$.

Hence we conclude that ω_{UA_2} results from ω_{VA_2} by setting $c = d = 0$ and notice further the similarity to e), that is $\omega_{f_{ij}} = -\omega_{e_{ij}}$.

i) $\mathbf{V_q(B_2)}$ vs. $\mathbf{U_q(B_2)}$

As indicated in a previous section we will now take a different Cartan matrix into account, namely B_2, see §5, Example (5.9). To avoid the listing of yet another large system of inequalities we refer to the paper [4] Example 4.4 as well as [8] for details. Here we follow the notation given in [4] and set

$$\omega = (\omega_{f_1}, \omega_{f_{12}}, \omega_{f_{122}}, \omega_{f_2}\omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_1}, \omega_{e_{12}}, \omega_{e_{122}}, \omega_{e_2}).$$

The weight vectors that will lead to $V_q(B_2)$ resp. $U_q(B_2)$ being graded are

$$\omega_{VB_2} = (-u+s,\ s,\ s+u,\ u,\ 0,\ t+u,\ 0,\ t+u,\ u-s,\ t+u-s,\ 2t+u-s,\ t)$$

respectively

$$\omega_{UB_2} = (-u+s,\ s,\ s+u,\ u,\ 0,\ 0,\ 0,\ 0,\ u-s,\ -s, -u-s, -u).$$

Again we register ω_{UB_2} as a special case of ω_{VB_2} by setting $t = -u$.

— *A best almost Grading* —

For this subsection we switch for once to the commutative case and set $R := \mathbb{K}[x_1, \ldots, x_n]$, where $\mathbb{K}$ is a commutative field.
As we have seen in some examples above, it is not always possible to compute weights, which will lead to a non-trivial grading of a given polynomial f. Hence one might ask, what is the *best approximation* of a grading?
More precisely, given a commutative polynomial

$$f = \sum_{i=1}^{k} c_i m_i \in R \tag{5}$$

with $c_i \in \mathbb{K} \setminus \{0\}$ and $m_i \in Mon_n(R)$ we wish to find the best *almost grading* of f, that is the highest graded part of f which has maximal length among all other gradings. Here we define the length of a polynomial f to be the number of contained terms in f, i.e. for f in (5) we have $lenght(f) = k$.

(6.6) Example
Let $R = \mathbb{R}[x, y]$ and consider $f = x^4 + y^5 + xy^4$. As the system $\{4\omega_x = 5\omega_y, 4\omega_x = \omega_x + 4\omega_y, 5\omega_y = \omega_x + 4\omega_y\}$ which is equivalent to $\{4\omega_x = 5\omega_y, 3\omega_x = 4\omega_y, \omega_y = \omega_x\}$ has the only solution $\omega_x = \omega_y = 0$ we look for a almost grading of length ≤ 2.
Obviously we can find weights to grade $x^4 + y^5$, for instance $\omega_x = 4t, \omega_y = 5t$, where t is an arbitrary parameter. For $x^4 + y^5$ to be the highest graded part we also have to take the inequalities $4\omega_x \geq \omega_x + 4\omega_y$ and $5\omega_y \geq \omega_x + 4\omega_y$ into account and obtain the solution $(\omega_x, \omega_y) = (t, 4/5t)$, with $t < 0$.
Of course we could also consider $x^4 + xy^4$ as highest graded part or $y^5 + xy^4$. We conclude that the best almost grading is not unique in general and notice that we are faced again with geometric questions as in the previous subsection.

The goal of this subsection is to formulate an algorithm, that produces for a polynomial $f \in R = \mathbb{K}[x_1, \dots, x_n]$ all best almost gradings that have length greater or equal to 2. We first recall the definition of the *Newton diagram* $\mathcal{N}(f)$ of $f = \sum_{\alpha \in \mathbb{N}} c_\alpha x^\alpha$, i.e.

$$\mathcal{N}(f) := \{\alpha \in \mathbb{N}_0^n \mid c_\alpha \neq 0\}.$$

By using this notation for a monomial $m = x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and setting $\omega = (\omega_1, \dots, \omega_n)$, we can express the induced equations for a grading of $m_1 + m_2$ with $m_1, m_2 \in Mon_n(R)$ by

$$\langle \mathcal{N}(m_1), \omega \rangle = \langle \mathcal{N}(m_2), \omega \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product.

The idea of the algorithm is as follows:
Compute as usual the system of equations S, which can then be used to decide whether f can be graded or not. In case f can be graded as a whole, we can calculate the corresponding weights as usual and are done. If the only possible grading for f is the trivial one, i.e. $\omega = 0$, we can use the set S to find step by step those equations which abort the full grading of f. Here S yields $|S|$ new sets $S'_1, \dots, S'_{|S|}$, each of which results from S by leaving out the i-th equation and hence have a cardinality of $|S| - 1$ each. The new systems S'_i are then again solved and tested for a non-trivial solution. Each system S'_i with a non-trivial solution yields a grading of maximal length among all other gradings and is hence a candidate for a best almost grading. For this we have to check whether and if so under which conditions this grading is of highest degree. This is done by taking the corresponding inequalities similar to Example 6.6 into account.

(6.7) Remark
Since the best almost grading of a polynomial f is not unique in general, we will use the notation $gr_\omega(f)$ to describe those terms of f which will be the best almost grading with respect to ω. For example, for $f = x^4 + y^5 + xy^4$ and $\omega = (\omega_x, \omega_y) = (t, 4/5t)$, where $t < 0$, we have $gr_\omega(f) = x^4 + y^5$. In the algorithm we will also use the notation $gr_{S'}(f)$. Furthermore we will denote the resulting best almost grading $x^t + y^t + x^t y^t$ by $bag_\omega(f)$ and the corresponding highest graded part $x^t + y^t$ by $hgp_\omega(f)$. In particular, for $gr_\omega(f) = \sum_\beta c_\beta x_1^{\beta_1} \cdots x_n^{\beta_n}$ we have $hgp_\omega(f) = \sum_\beta c_\beta x_1^{\beta_1 \omega_1} \cdots x_n^{\beta_n \omega_n}$.

Let us have a look at the algorithm. Without loss of generality and for simplicity we assume all terms of f to be monic and not constant.

Algorithm 3 Computation of all possible highest graded parts of f of lenght ≥ 2.

```
procedure HGP(Polynomial $f$ in variables $x_1 \dots, x_n$)
    $M := Mon(f)$;                                         ▷ contains all terms of $f$
    $S := \emptyset$, $W := Vector(\omega_1, \dots, \omega_n)$;      ▷ Weights for the variables $x_1 \dots, x_n$
    for $(m_1, m_2) \in M \times M$ with $m_1 \neq m_2$ do
        $S := S \cup \{\langle \mathcal{N}(m_1), W\rangle = \langle \mathcal{N}(m_2), W\rangle\}$;
    end for
    $\omega^* :=$ solution of $S$;                              ▷ $\omega^* = (\omega_1^*, \dots, \omega_n^*)$
    if $\omega^* \neq 0$ then
        return $bag_\omega(f)$;                            ▷ the grading of $f$ by $\omega^*$.
    else
        $SL := [], Candidates := []$;
        if $|S| = 1$ then
            return The highest graded part of $f$ has lenght 1.
        end if
        for $eq \in S$ do
            add the set $S' := S \setminus \{eq\}$ to the list $SL$;
            $\omega^*_{S'}$ := solution of $S'$;
            if $\omega^*_{S'} \neq 0$ then
                $gr_{S'}(f)$ := corresponding terms of $f$ that can be graded;
                add the tuple $C_{S'} := (S', \omega^*_{S'}, gr_{S'}(f))$ to the list $Candidates$;
            end if
        end for
        $HG := []$;
        for $C_{S'} \in Candidates$ do
            $V := Mon(gr_{S'}(f))$, $NV := Mon(f) \setminus V$;
            for $(m_1, m_2) \in V \times NV$ do
                $S' := S' \cup \{\langle \mathcal{N}(m_1), W\rangle \geq \langle \mathcal{N}(m_2), W\rangle\}$;
            end for
            $\omega^*_{S'}$ := solution of $S'$;
            if $\omega^*_{S'} \neq 0$ then
                $bag_{S'}(f)$ := best almost grading of $f$ w.r.t. $S'$.
                add $C_{S'} := (S', \omega^*_{S'}, gr_{S'}(f), bag_{S'}(f), hgp_{S'}(f))$ to the list $HG$;
            end if
        end for
        while $HG = []$ do
            repeat lines 12-34 for each $S'$ in $SL$;     ▷ that is replace $S$ in above by $S'$
        end while
    end if
    return $HG$;
end procedure
```

Termination:
The algorithm terminates due to lines 35 - 37. More precisely, if f is a monomial or even a sum of two monomials then the termination follows as f can be graded and otherwise we can assume $|S| \geq 2$. With each pass of the while-loop the cardinality of the sets S' decreases by one and hence at least when HG is still empty while all S' contain only one equation the if query in line 12 terminates the algorithm.
Correctness:
The correctness of the algorithm is provided by the fact, that, whenever f cannot be graded, we start with the set S and always compute a grading of maximal length first in lines 18 - 21 and afterwards check in lines 24 - 34 if this can also be of highest degree among all other terms. If no such grading exists, we will display that there are only those possible of length 1.

An implementation of the algorithm in MAPLE and applying it to Example (6.6) yields

(ω_x, ω_y)	$gr_\omega(f)$	$bag_\omega(f)$	$hgp_\omega(f)$
$(t, \frac{4t}{5}),\ t < 0$	$x^4 + y^5$	$x^{4t} + y^{4t} + x^t y^{4t}$	$x^{4t} + y^{4t}$
$(t, \frac{3t}{4}),\ t > 0$	$x^4 + xy^4$	$x^{4t} + y^{\frac{15t}{4}} + x^t y^{3t}$	$x^{4t} + x^t y^{3t}$
$(t, t), t > 0$	$y^5 + xy^4$	$x^{4t} + y^{5t} + x^t y^{4t}$	$y^{5t} + x^t y^{4t}$

We deduce from this easy example, that with an increase in the number of considered terms and/or variables in a polynomial the problem at hand gets complicated very quickly.

(6.8) Remark
Besides the fact, that we can compute a best almost grading of a polynomial f, we also obtain information about how long it is, i.e. $lenght(gr_\omega(f))$.

We will continue our calculations in the following section, however in a different context. Simply speaking we will not focus on gradings any longer but rather specifically designed solutions for integer programming problems which arise from the relations of a non-commutative algebra.

§ 7 Balancing of Weights

Given an algebra $A = \mathbb{K}\langle x_1, \ldots, x_n \mid \{x_j x_i = c_{ij} x_i x_j + d_{ij}\}_{1 \leq i < j \leq n}\rangle$ we have seen in Theorem (3.11) how to interpret the problem of deciding whether a weighted monomial ordering $\prec_\omega$ is admissible is equivalent to solving the system of inequalities

$$\omega_i + \omega_j \geq \max_l \{deg_\omega(d_{ijl})\} \tag{6}$$

for $1 \leq i < j \leq n$ and each l. In this section we are going to elaborate on this topic by analyzing the structure of ω. More precisely we focus on algorithms that pick out *balanced* weight vectors, where we have to specify the meaning of balance.

— *Alternative Solutions* —

As mentioned in a previous section, the broad range of well known algorithms to solve linear, reps. integer programming problems lack the output of possibly alternative solutions. As this is an important part in order to find specifically designed solutions though we will discuss a few methods to attack this issue.

In their paper "Finding multiple solutions to general integer linear programs" [18] Tsai, Lin and Hu proposed a method of how to find alternative *optimal* solutions for a given integer program. The basic idea is to compute an optimal solution (if it exists) using an appropriate algorithm for the given problem and then cut out this solution by extending the constraints such that the computed solution is no longer part of the feasible region. Furthermore another constraint is added which fixes the value of the objective function to the value of the optimal solution. Solving this new problem then yields an alternative optimal solution to the first one.
Considering the problem (6), we are not searching for alternative *optimal* but rather alternative *feasible* solutions. Hence the following part slightly differs compared to their treatment of the problem.
The first step is to introduce the new constraints, that cut out a given solution of the feasible region:
Say $\omega^* = (\omega_1^*, \ldots, \omega_n^*)$ is a solution to (6). Adding the constraint

$$\sum_{k=0}^{n} |\omega_k - \omega_k^*| \geq 1$$

to the set of the given constraints in (6) then cuts out the interior of a n-dimensional cube with edge length 1 and center ω^*. However, this is no linear constraint, hence it has to be linearized to match our model.

(7.1) Proposition ([18], Proposition 2)
Let $\alpha_j \in \{0,1\}, W_j \geq 0$ and M is a large constant, then

$$\sum_{j=0}^{n} |\omega_j - \omega_j^*| \geq 1 \;\Leftrightarrow \begin{cases} (i) & 0 \leq W_j - \omega_j + \omega_j^* \leq M \cdot (1-\alpha_j),\; j = 1,\dots,n \\ (ii) & 0 \leq W_j - \omega_j^* + \omega_j \leq M \cdot \alpha_j,\; j = 1,\dots,n \\ (iii) & \sum_{j=1}^{n} W_j \geq 1. \end{cases}$$

Proof
First let us prove " $\Leftarrow$ ":
Let $(i), (ii)$ and (iii) be given.
If $\omega_j - \omega_j^* > 0$ for some j, then (ii) implies $\alpha_j = 1$ and hence by (i) it follows

$$|\omega_j - \omega_j^*| = W_j.$$

If $\omega_j - \omega_j^* < 0$ for some j, then (i) implies $\alpha_j = 0$ and hence by (ii) it follows

$$|\omega_j - \omega_j^*| = W_j.$$

Additionally if $\omega_j = \omega_j^*$ for some j then (i) and (ii) imply $W_j = 0$, that is $|\omega_j - \omega_j^*| = W_j$. Summarizing the above yields $\sum_{j=0}^{n} |\omega_j - \omega_j^*| = \sum_{j=1}^{n} W_j$ and in particular

$$\sum_{j=0}^{n} |\omega_j - \omega_j^*| \geq 1 \Leftrightarrow \sum_{j=1}^{n} W_j \geq 1.$$

The other direction " $\Rightarrow$ " follows in the same matter since $\sum_{j=0}^{n} |\omega_j - \omega_j^*| \geq 1$ implies that there is some j, such that $\omega_j \neq \omega_j^*$, hence W_j and α_j can be chosen accordingly to satisfy $(i), (ii)$ and (iii). $\square$

(7.2) Remark
Linearizing the inequality $\sum_{j=0}^{n} |\omega_j - \omega_j^*| \geq 1$ in this way yields $4n + 2$ additional constraints and requires $2n$ more variables, whereas the feasible region decreases insignificantly in general. Therefore the runtime per iteration of the algorithm increases with each step.

In the following we will present a pseudo-code of the algorithm in question. Note, that we do not fix the objective value of the optimal solution and hence search for feasible, not necessarily optimal solutions. Given an integer programming problem with objective function f and a set of constraints C, we will now present a basic version of the algorithm. Note that we bound the maximal number of computed alternative solutions by an integer $runs \in \mathbb{N}$. It also assures the termination of the algorithm whereas the correctness is provided by Proposition 7.1.

Algorithm 4 Basic Algorithm for Alternative Solutions

```
1: procedure AltSol($f$, $C$ :: set, runs :: posint)
2:     $\omega^*$ :=optimal solution of the original IP with obj. function $f$ and constraints $C$.
3:     r:= 1
4:     M»0                                        ▷ Set a value for $M$ in Prop.7.1
5:     S:= $\emptyset$
6:     while $r \leq runs$ do
7:         $S := S \cup \{0 \leq W_j + \omega_j - \omega_j^* \mid j = 1, \dots, n\}$     ▷ Add the new constraints
8:         $S := S \cup \{W_j + \omega_i - \omega_i^* \leq M \cdot a_j \mid j = 1, \dots, n\}$
9:         $S := S \cup \{0 \leq W_j + \omega_j^* - \omega_j \mid j = 1, \dots, n\}$
10:        $S := S \cup \{W_j + \omega_j^* - \omega_j \leq M \cdot (1 - a_j) \mid j = 1, \dots, n\}$
11:        $S := S \cup \{1 \leq \sum_{z=1}^{n} W_j\}$
12:        $S := S \cup \{W_j \geq 0 \mid j = 1, \dots, n\}$
13:        $NewC := C \cup S$;
14:        $\omega^*$ :=optimal solution of the IP with obj. function $f$ and constraints $NewC$,
15:                where the $a_j$ are set as binary variables.
16:        r:= r+1;
17:        output: $\omega^*$ and objective value.
18:    end while
19: end procedure
```

However the performance of the algorithm strongly relies on the performance of the algorithm chosen to solve the arising integer programming problems. So far we have not talked about the different techniques for solving linear, resp. integer programming problems but since we would like to apply Proposition (7.1) to various examples, a few words are in order.

The classical approach to linear programming problems, i.e. the Simplex algorithm, was given by Dantzig [7]. However it is not directly applicable for the problem at hand, as we focus on integer programming problems. Further methods include the interior point algorithm by Karmarkar or the big M method which is a modification of the simplex algorithm.

We are going to use a specifically designed technique for integer programming problems, i.e. the *branch and bound* method which will be outlined in the following subsection.

— *Branch and Bound* —

The idea of the branch and bound method is fairly simple. We will explain the basic steps first and go into detail by examining a two-dimensional example afterwards.
One first solves the *relaxation* of the integer programming problem, that is we assume the decision variables to be elements of $\mathbb{R}$ instead of $\mathbb{Z}$, which yields a *linear* programming problem (LP). If the solution $(\omega_1^*, \dots, \omega_n^*)$ is integer-valued, it coincides with the solution of the IP and we are done. If the solution is not integer-valued, that is $\exists\ i$ with $\omega_i^* \notin \mathbb{Z}$, we will split the IP into two carefully chosen LP's, that is we split the feasible region into two disjoined regions, neither of which contains $(\omega_1^*, \dots, \omega_n^*)$. Each of these regions corresponds to a new LP, which arises from the original LP by adding new constraints, i.e. we have a *branching*. We actually branch on a specific ω_i. The occurring LP's are then again solved. If an arising solution is integer-valued, it is a candidate for the optimal solution of the original IP and no further branching is needed, i.e. it *bounds* the domain of ω_i to be regarded in this branching. If not, we have another branching and the process repeats itself. At some point all possible cases were considered and we get an optimal solution.

Let us now demonstrate the process along a two-dimensional example. Given an integer programming problem (IP), for example

$$\begin{aligned} \text{Max} \quad & \omega_1 + \omega_2 \\ \text{s.t.} \quad & 2\omega_1 + \omega_2 \le 11, \\ & \omega_1 + 5\omega_2 \le 21, \\ & \omega_1, \omega_2 \ge 0, \\ & \omega_1, \omega_2 \in \mathbb{Z}, \end{aligned}$$

we first compute the solution of the relaxation using the Simplex algorithm and obtain $(\omega_1^*, \omega_2^*) = (\frac{34}{9}, \frac{31}{9}) \notin \mathbb{Z}^2$.

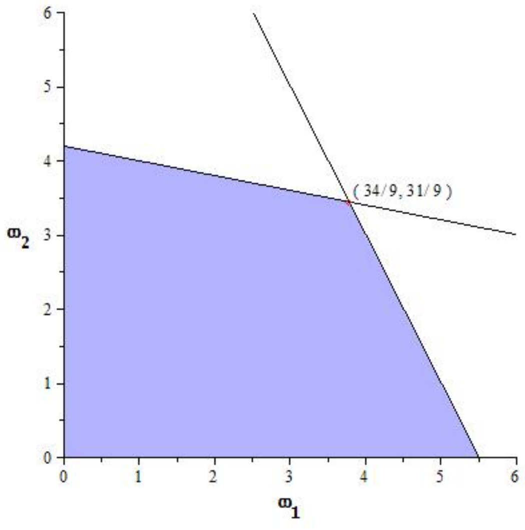

Figure 6: Feasible region of the original IP

Since the solution is not integer-valued, we have to choose a variable to branch on, say ω_1. We will divide the feasible region into two disjoint regions by adding new constraints, namely $\omega_1 \leq \lfloor \frac{34}{9} \rfloor = 3$ and $\omega_1 \geq \lceil \frac{34}{9} \rceil = 4$ that is we consider two new LP's :

$$
\begin{array}{lll}
LP_1 : & \text{Max} & \omega_1 + \omega_2 \\
& \text{s.t.} & 2\omega_1 + \omega_2 \leq 11, \\
& & \omega_1 + 5\omega_2 \leq 21, \\
& & \omega_1, \omega_2 \geq 0 \\
& & \omega_1 \leq 3 \\
& & \omega_1, \omega_2 \in \mathbb{R}
\end{array}
$$

and

$$
\begin{array}{lll}
LP_2 : & \text{Max} & \omega_1 + \omega_2 \\
& \text{s.t.} & 2\omega_1 + \omega_2 \leq 11, \\
& & \omega_1 + 5\omega_2 \leq 21, \\
& & \omega_1, \omega_2 \geq 0 \\
& & \omega_1 \geq 4 \\
& & \omega_1, \omega_2 \in \mathbb{R}.
\end{array}
$$

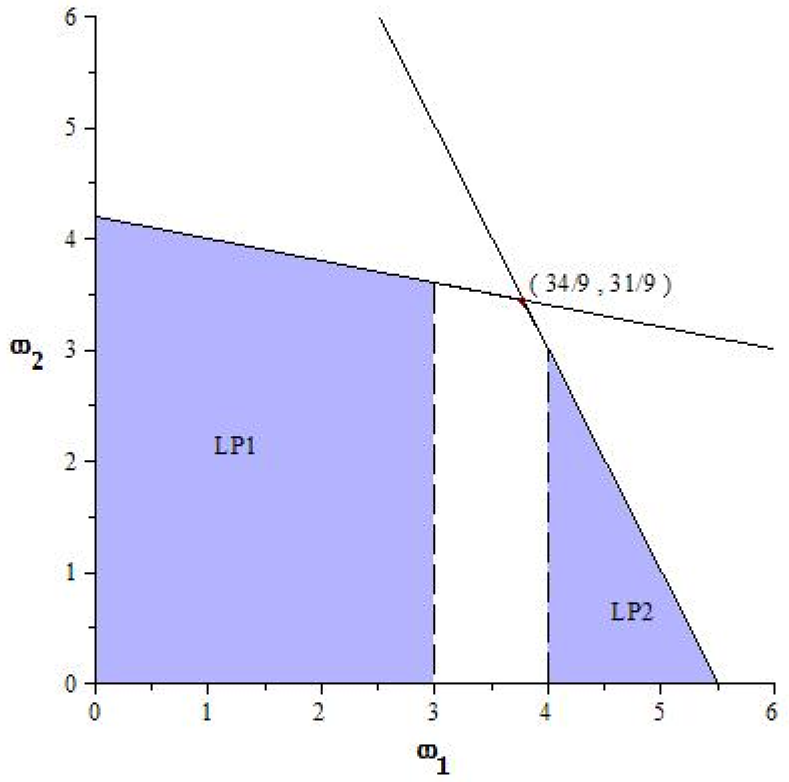

Figure 7: Decomposition of the original feasible region

Solving the LP's yield the solutions $(3, \frac{18}{5})$ and $(4, 3)$ respectively. We see that the second solution $(4, 3)$ with objective value 7 is integer-valued and hence no further branching is needed since adding even more constraints cannot result in finding a better solution. The solution $(\omega_1, \omega_2) = (3, 4)$ is hence a candidate for the optimal solution. Considering the first solution we note, that $\omega_1 = 3 \in \mathbb{Z}$ and we cannot branch on ω_1 again. Hence we branch on ω_2 and obtain analogously to the previous step the LP's :

$$\begin{array}{lrl}
\text{Max} & \omega_1 + \ \omega_2 & \\
\text{s.t.} & 2\omega_1 + \ \omega_2 & \leq 11, \\
& \omega_1 + 5\omega_2 & \leq 21, \\
& \omega_1, \omega_2 & \geq 0 \\
& \omega_1 & \leq 3 \\
& \omega_2 & \leq 3 \\
& \omega_1, \omega_2 & \in \mathbb{R}
\end{array}$$

and

$$\begin{array}{rrl} \text{Max} & \omega_1 + \ \omega_2 & \\ \text{s.t.} & 2\omega_1 + \ \omega_2 & \leq 11, \\ & \omega_1 + 5\omega_2 & \leq 21, \\ & \omega_1, \omega_2 & \geq 0 \\ & \omega_1 & \leq 3 \\ & \omega_2 & \geq 4 \\ & \omega_1, \omega_2 & \in \mathbb{R} \end{array}$$

which yield the solutions $(3, 3)$ and $(1, 4)$ with objective value 6 resp. 5. Note, that this second step was unnecessary in fact since the resulting solutions, that is the objective values, were bounded by $3 + \frac{18}{5}$ which is already smaller than 7. We displayed it however to demonstrate the idea of the method.
Furthermore the process is completed and we see that $(\omega_1, \omega_2) = (3, 4)$ is the optimal integer solution.
Additionally one might want to take the solution of the relaxation of the original IP and round it to get an integer-valued solution. Although it would work in this example, it is only by accident and is not successful in general. Rounding might even yield an infeasible solution.

However if the feasible region is unbounded, this method may not terminate. Note, that the feasible regions we have considered so far in our examples of the previous sections were unbounded. We can easily solve this problem in our case however, since we are looking for specifically designed solutions and hence artificially bound the intervals of the decision variables and therefore bound the feasible region.

— *Finding the Balance* —

Now it is time to link all of the above and apply the algorithm, which translates Proposition (7.1) and yields alternative solutions to an integer programming problem. We used MAPLE to implement the algorithm. Note that the `LPSolve` command of the `Optimization` package in MAPLE uses the branch and bound method. One can set the maximum range of branches per iteration by setting a value for the option $depthlimit = n$.
Furthermore note that the termination of the algorithm is either determined by $runs$, the maximal number of alternative solutions to be computed or by the `LPSolve` command when a constructed integer programming problem in an iteration is infeasible. In the latter case an error message “no feasible solution found” will be displayed. Moreover, the option *depthlimit* in the `LPSolve` command plays a role in finding a solution as

described in the subsection *Branch and Bound* as well as in the context of the runtime of the algorithm. Here we suggest the reader to test different parameters to get a good sense for it. Additionally as most problems considered above are unbounded we chose to control the termination of the algorithm by the parameter *runs*.

(7.3) Example
To “assure” the algorithm works properly, we consider as a first example the corresponding IP of the guitar manufacturing company from the beginning of section 2, that is

$$\begin{array}{lrcl} \text{Max} & 220p_e + 308p_a & & \\ \text{s.t.} & p_e + \quad p_a & \leq & 12 \\ & 3p_e \quad + 6p_a & \leq & 51. \end{array}$$

We obtain the solutions $(7,5), (8,4), (5,6), \ldots$ along with a descending chain of objective values.

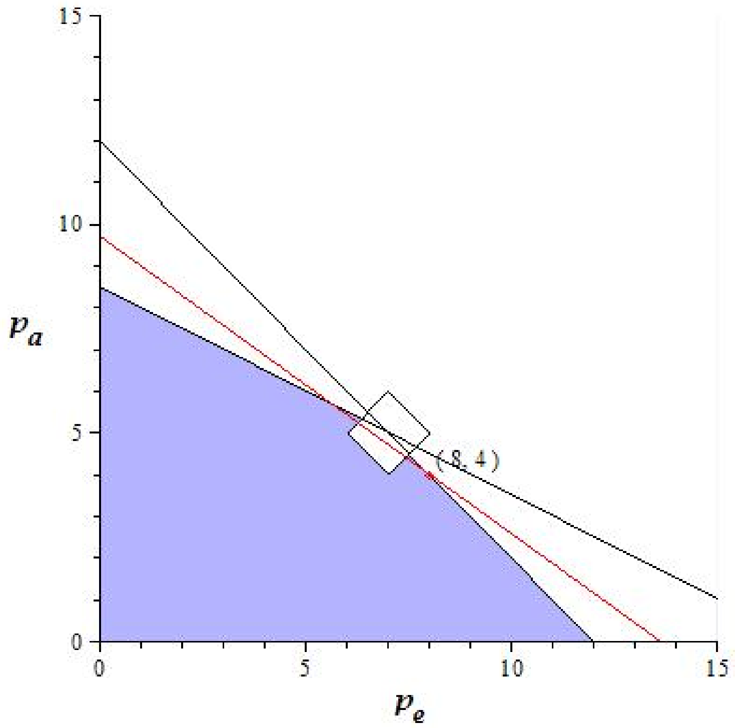

Figure 8: Feasible region for alternative solutions

The graphic shows the the first computed alternative solution $(8,4)$, after the cube of edge length 1 and interior $(7,5)$ has been cut out.

We will discuss how to use the algorithm to point it in the direction of computing possibly balanced solutions. Note that we do not require the IP to have multiple optimal solutions. Whether the i-th alternative solution is optimal or not can easily be deduced from comparing its objective value with the objective value of the first solution given by the algorithm.
Now we wish to apply the algorithm to the theory presented in the previous sections. However we will not use the basic algorithm, but focus on different additional features depending on the structural requirements of our desired solution. So far we have basically dealt, in the context of G-algebras, with computing weight vectors that will lead to an admissible weighted monomial ordering, filtrations and the (quantum) universal enveloping algebra of a Lie algebra. From the viewpoint of *balancing weights* we will focus in the following on

1) predefined regions
2) symmetry (of certain blocks) and
3) minimal upper boundaries.

By "predefined regions" we mean setting *a priori boundaries* thereby avoiding the inconvenience of an unbounded integer programming problem. One can either add further constraints to the set C in Algorithm 3 or build a procedure into the algorithm which sets boundaries for all or certain blocks of the decision variables, thereby forcing the algorithm to look for solutions in a predefined area which may intersect the feasible region. Here the output of alternative solutions comes in quite handy as we can set the boundaries rather loosely and do not have to alter the set borders very often in case these were too strict and led to infeasibility.
Additionally the `LPSolve` command in MAPLE allows us to set an initial point influencing where to start the computation. A *balanced* initial point may hence be helpful.

Let us now take on an example which is more adjusted to our setting.

(7.4) Example
We consider again the quantum enveloping algebra $U_q(A_2)$ and, of course, the connection to $V_q(A_2)$. As before, we regard Yamane's bases for these algebras (see [19]), i.e. for $U_q(A_2)$ we have

$$\{f^\gamma k^\beta e^\alpha \mid \alpha,\gamma \in \mathbb{N}^3, \beta \in \mathbb{Z}^2\},$$

where $e^\alpha = e_{12}^{\alpha_1} e_{13}^{\alpha_2} e_{23}^{\alpha_3}, k^\beta = k_1^{\beta_1} k_2^{\beta_2}$ and $f^\gamma = f_{12}^{\gamma_1} f_{13}^{\gamma_2} f_{23}^{\gamma_3}$. In [4] the need for a weight vector

$$\omega = (\omega_{f_{12}}, \omega_{f_{13}}, \omega_{f_{23}}, \omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_{12}}, \omega_{e_{13}}, \omega_{e_{23}})$$

for $V_q(A_2)$ is given that makes $V_q(A_2)$ a G-algebra, which can be computed as a solution of the corresponding integer programming problem, where we minimize the objective function:

$$f(\omega) = \omega_{f_{12}} + \omega_{f_{13}} + \omega_{f_{23}} + \omega_{k_1} + \omega_{k_2} + \omega_{l_1} + \omega_{l_2} + \omega_{e_{12}} + \omega_{e_{13}} + \omega_{e_{23}}$$

subject to the constraints

$$\begin{array}{cccc}
\omega_{e_{12}} \geq 1, & \omega_{e_{13}} \geq 1, & \omega_{e_{23}} \geq 1, & \\
\omega_{f_{12}} \geq 1, & \omega_{f_{13}} \geq 1, & \omega_{f_{23}} \geq 1, & \\
\omega_{k_1} \geq 1, & \omega_{k_2} \geq 1, & \omega_{l_1} \geq 1, & \omega_{l_2} \geq 1,
\end{array}$$

$$\begin{array}{rr}
-\omega_{e_{12}} + \omega_{e_{13}} - \omega_{e_{23}} \leq -1, & -\omega_{f_{12}} + \omega_{f_{13}} - \omega_{f_{23}} \leq -1, \\
2\omega_{k_1} - \omega_{e_{12}} - \omega_{f_{12}} \leq -1, & 2\omega_{k_2} - \omega_{e_{23}} - \omega_{f_{23}} \leq -1, \\
2\omega_{l_1} - \omega_{e_{12}} - \omega_{f_{12}} \leq -1, & 2\omega_{l_2} - \omega_{e_{23}} - \omega_{f_{23}} \leq -1, \\
\omega_{f_{12}} + 2\omega_{l_2} - \omega_{e_{23}} - \omega_{f_{13}} \leq -1, & 2\omega_{k_1} + \omega_{f_{23}} - \omega_{e_{12}} - \omega_{f_{13}} \leq -1, \\
2\omega_{k_1} + 2\omega_{k_2} - \omega_{e_{13}} - \omega_{f_{13}} \leq -1, & 2\omega_{k_2} + \omega_{e_{12}} - \omega_{e_{13}} - \omega_{f_{23}} \leq -1, \\
2\omega_{l_1} + \omega_{e_{23}} - \omega_{e_{13}} - \omega_{f_{12}} \leq -1, & 2\omega_{l_1} + 2\omega_{l_2} - \omega_{e_{13}} - \omega_{f_{13}} \leq -1,
\end{array}$$

see Theorem (3.11) for its construction.
In addition to their solution

$$\begin{aligned}
\omega &= (\omega_{f_{12}}, \omega_{f_{13}}, \omega_{f_{23}}, \omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_{12}}, \omega_{e_{13}}, \omega_{e_{23}}) \\
&= (3, 5, 3, 1, 1, 1, 1, 1, 1, 1)
\end{aligned}$$

our algorithm yields among other the alternative solutions

$$\begin{array}{rl}
& (3, 4, 2, 1, 1, 1, 1, 1, 2, 2), \\
& (2, 4, 3, 1, 1, 1, 1, 2, 2, 1), \\
\text{or} & (2, 3, 2, 1, 1, 1, 1, 2, 3, 2).
\end{array}$$

Furthermore, by solving the corresponding integer programming problem for $U_q(A_2)$, that is we replace the inequalities $\omega_{l_i} \geq 1$ by $\omega_{l_i} = -\omega_{k_i}$ for $i = 1, 2$, we obtain for example the solution

$$(2, 3, 2, 1, 1, -1, -1, 2, 3, 2).$$

What can we conclude so far?

1) The search for alternative solutions enables the possibility for specifically designed solutions, for example symmetric ones as we have seen in the previous example.

2) Additionally we can set borders for the variables, that is we can determine the smallest $c \in \mathbb{N}$ such that $\omega_i \leq c$ for all i, that is the minimal upper boundary as mentioned above.

The second point actually arises from another issue. Imagine the only optimal solution of an IP takes the form $\omega = (149, 4, 11, 6)$. Here the approach of a priori restrictions for the variables may turns out tedious. Hence regarding this solution serves as a starting point. Taking the usual objective function we note that the objective value has to be greater than or equal to $149 + 4 + 11 + 6 = 170$. Therefore trying to find a balanced solution by lowering the value for ω_1 by $d \in \mathbb{N}$ results in raising the values of ω_2, ω_3 and ω_4 in total at least by d, which gives us a better sense of where to look for balanced solutions. Of course the algorithm cannot construct balanced solutions if there are none, hence this has to be viewed as an approximation towards a balance. In such a case we can build a *while-loop* into the algorithm which lowers the upper boundary for ω_1 step by step using the constraints as long as the resulting IP's remain feasible.

For example:

```
while "IP feasible" do
    $m$ :=max($\omega^*$)
    $am$ :=argmax($\omega^*$)
    $NewC := NewC \cup \{\omega_{am} \leq m - 1\}$;
    $\omega^*$ :=optimal solution of the IP with obj. function $f$ and constraints $NewC$,
          where the $a_j$ are set as binary variables.
    output: $\omega^*$ and objective value.
end while
```

Hence we are able to extend the theory from section 3 in the way that we are not only capable to compute any solution, but one with a certain structure that can also be applied in the context of filtrations in section 4. We have already seen weight vectors which can be used for a weighted degree filtration in the previous example. Furthermore, although we did not mention it before, the weight vector ω corresponding to $V_q(A_2)$, i.e. $(2, 3, 2, 1, 1, 1, 1, 2, 3, 2)$ is bounded by 3 in the meaning of point 2) above. More precisely there exists no solution ω such that $\omega_i \leq 2$ for all i. Regarding $U_q(A_2)$ however, we can consider two cases

i) The first one, namely $\omega_{k_i} = -\omega_{l_i} \neq 0$, we have already taken into account above, i.e. $\omega = (2, 3, 2, 1, 1, -1, -1, 2, 3, 2)$. Again the solution is bounded by 3.

ii) The second case is $\omega_{k_i} = \omega_{l_i} = 0$ and here we are able to lower the boundary and obtain the weight vector $\omega = (1, 1, 1, 0, 0, 0, 0, 1, 1, 1)$. Obviously we can not achieve a lower boundary.

Let's regard B_2 as underlying Cartan matrix again.

(7.5) Example
We refer the reader to [4], Example 4.4 for details of the relations and will proceed in a manner analogue to the previous example and set

$$\omega = (\omega_{f_1}, \omega_{f_{12}}, \omega_{f_{122}}, \omega_{f_2}\omega_{k_1}, \omega_{k_2}, \omega_{l_1}, \omega_{l_2}, \omega_{e_1}, \omega_{e_{12}}, \omega_{e_{122}}, \omega_{e_2}).$$

In [4] the authors obtained the weight vector

$$\omega = (1, 1, 2, 2, 1, 1, 1, 1, 4, 9, 15, 7)$$

which is neither balanced nor symmetric in any form and hence yields a good starting point for the application of our algorithm. First we focused on finding a symmetric solution while keeping the upper boundary as small as possible. Here we obtained the weight vector

$$\omega = (3,\ 6,\ 10,\ 5,\ 1,\ 1,\ 1,\ 1,\ 3,\ 6,\ 10,\ 5).$$

Moreover, we are able to lower the boundary from 10 to 9 by leaving out the symmetry condition and get among other the solution

$$\omega = (3,\ 5,\ 8,\ 4,\ 1,\ 1,\ 1,\ 1,\ 2,\ 5,\ 9,\ 5).$$

Considering the two cases for $U_q(B_2)$, we obtain the following:

i) $\omega_{k_i} = -\omega_{l_i} \neq 0$:
Again we get the same solution as for $V_q(B_2)$ where only the signs of ω_{l_i} change from positive to negative, i.e. $\omega = (3,\ 6,\ 10,\ 5,\ 1,\ 1, -1, -1,\ 3,\ 6,\ 10,\ 5)$, if we want to keep the symmetry and $\omega = (3,\ 5,\ 8,\ 4,\ 1,\ 1, -1, -1,\ 2,\ 5,\ 9,\ 5)$ if not, yielding the same boundary as for $V_q(A_2)$.

ii) $\omega_{k_i} = \omega_{l_i} = 0$:
Here we can lower the upper boundary even further and obtain the symmetric solution
$$\omega = (2, 3, 5, 3, 0, 0, 0, 0, 2, 3, 5, 3).$$
Moreover leaving out the symmetry condition does not result in finding a lower boundary and hence the solution is bounded by 5.

Furthermore, note that all of the computed solutions were also optimal ones.

(7.6) Observation
Examining the last examples we have seen an idea of the word *balance*, namely the decomposition of the weight vector ω into different blocks which are bounded for themselves and possibly symmetric to each other.

(7.7) Remark
In terms of elimination of variables for G-algebras described in section 3, subsection 3, we wish to point out, that the existence of elimination orderings can easily be checked by replacing each inequality of the form $b^T\omega \leq -1$ by $b^T\omega \leq 0$ and $\omega_j \geq 1$ by $\omega_j = 0$ for the variables *not* to be eliminated. Of course the theory and algorithm described above can be applied here to find balanced weight vectors for elimination orderings.

Considering $V_q(A_2)$ or $U_q(A_2)$ for example we deduce, that there exist elimination orderings in the case we want to eliminate $\{e_{12}, e_{13}, e_{23}\}$, $\{e_{12}, e_{13}\}$ or $\{e_{13}, e_{23}\}$ but the elimination of e_{12} and e_{23} is not possible. As one might suspect, the same holds for the f_{ij}. Moreover, we have already seen a weight vector leading to the elimination of all the e_{ij} and f_{ij}, namely $\omega = (1,1,1,0,0,0,0,1,1,1)$.

Although we kept our computations of weights mainly within the context of a quantum universal enveloping algebra of a Lie algebra, we wish to stress that this was merely for demonstrating the features of the underlying algorithm. The theory however applies to general integer programming and hence might prove useful in various mathematical aspects.

(7.8) Observation
As we have seen, the computation of balanced - or more structured - weights is indeed possible, but in general could not be made into a completely automatic procedure. That is, one has to work with the concrete non-commutative algebra and combine algorithms with human insight.

Conclusions and Future Directions

In this thesis we have seen naturally arising connections mostly between non-commutative algebras and the theory of integer programming. We designed algorithms and techniques for finding balanced solutions for integer programming problems within the context of G-algebras as well as for gradings resp. best almost gradings.
We have shown various roles weight vectors play in regarding monomial orderings in G-algebras, however the developed methods in this work for the computation of these weight vectors apply to a more general context. Furthermore, our theory for balanced weights is only applicable for integer programming problems. i.e. in $\mathbb{Z}$. By cutting out a cube with edge length one and an integer interior point, we decrease the number of possibly integer solutions only by one, whereas in $\mathbb{Q}$ we actually cut out a whole "cube of solutions". Hence studying balanced solutions over $\mathbb{Q}$ is still an interesting question.
As an application of these weight vectors we considered a grading or filtration of a G-algebra.
Within the study of a grading for a non-commutative algebra

$$A = \langle x_1, \dots, x_n \mid \{x_j x_i = c_{ij} x_i x_j + d_{ij}\}_{1 \le i < j \le n} \rangle$$

we noticed a pattern resulting from the corresponding ideal of relations. In geometric terms, the relations of A in combination with the condition for an admissible ordering describe a feasible region for those weight vectors that induce an admissible weighted monomial ordering. The intersection of all borders of this region yields a geometric object which contains all weight vectors, that can be used to find a grading for A.
Here we mainly focused on quantum universal enveloping algebras of special linear Lie algebras for our calculations, as these yield interesting examples due to their complex structure. More precisely, in view of a balance, we were able to improve existing weights for $U_q(A_2)$ and $U_q(B_2)$, but our methods do not depend on the choice of the corresponding Cartan matrix. Furthermore, our calculations gave rise to a reasonable characterization of a balanced weight vector ω in the first place. For this one has to keep in mind, that only an approximation of balance is possible in general. What stood the test was finding tight boundaries for ω on one hand and on the other hand decomposing ω into several blocks which are possibly symmetric to each other.
While we shortly elaborated on elimination orderings as well, we think that among other an interesting application of our work lies in the study of the well-known Gröbner Walk. Here, the computation of weights is a vital aspect and the option of possibly balancing these weights might find a useful purpose.

References

[1] Daniel Andres. *Noncommutative Computer Algebra with Applications in Algebraic Analysis.* PhD thesis, RWTH Aachen University, 2013. `http://darwin.bth.rwth-aachen.de/opus3/volltexte/2014/4928/`.

[2] Joachim Apel. *Gröbnerbasen in nichtkommutativen Algebren und ihre Anwendung.* PhD thesis, 1988.

[3] Thomas Becker and Volker Weispfenning. *Gröbner bases.* Springer, 1993.

[4] José L Bueso, José Gómez-Torrecillas, and Francisco Javier Lobillo. Re-filtering and exactness of the Gelfand–Kirillov dimension. *Bulletin des sciences mathematiques*, 125(8):689–715, 2001.

[5] José Luis Bueso, José Gómez-Torrecillas, and Alain Verschoren. *Algorithmic methods in non-commutative algebra: Applications to quantum groups*, volume 17. Kluwer, 2003.

[6] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.

[7] George Bernard Dantzig. *Linear programming and extensions.* Princeton university press, 1998.

[8] Corrado De Concini and Claudio Procesi. *Quantum groups.* Springer, 1993.

[9] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann. Singular 4-0-2 — A computer algebra system for polynomial computations, 2015. `http://www.singular.uni-kl.de`.

[10] Jacques Dixmier. *Enveloping algebras*, volume 14. Newnes, 1977.

[11] Juan Ignacio Garcia Garcia, Jesus Garcia Miranda, and Francisco Javier Lobillo. Elimination orderings and localization in PBW algebras. *Linear Algebra and Its Applications*, 430(8):2133–2148, 2009.

[12] G.-M. Greuel, V. Levandovskyy, A. Motsak, and H. Schönemann. Plural — A singular 4.0 subsystem for computations with non-commutative polynomial algebras., 2015. `http://www.singular.uni-kl.de`.

[13] James E Humphreys. *Introduction to Lie algebras and representation theory*, volume 9. Springer Science & Business Media, 1972.

[14] Abdelilah Kandri-Rody and Volker Weispfenning. Non-commutative gröbner bases in algebras of solvable type. *Journal of Symbolic Computation*, 9(1):1–26, 1990.

[15] Viktor Levandovskyy. *Non-commutative computer algebra for polynomial algebras: Gröbner bases, applications and implementation.* PhD thesis, TU Kaiserslautern, 2005. `http://kluedo.ub.uni-kl.de/volltexte/2005/1883/`.

[16] Teo Mora. Gröbner bases in non-commutative algebras. In *Symbolic and algebraic computation*, pages 150–161. Springer, 1989.

[17] José Gómez Torrecillas. Gelfand-kirillov dimension of multi-filtered algebras. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 42(01):155–168, 1999.

[18] Jung-Fa Tsai, Ming-Hua Lin, and Yi-Chung Hu. Finding multiple solutions to general integer linear programs. *European Journal of Operational Research*, 184(2):802–809, 2008.

[19] Hiroyuki Yamane. A Poincaré-Birkhoff-Witt Theorem for Quantized Universal Enveloping Algebras of Type An. *Publications of the Research Institute for Mathematical Sciences*, 25(3):503–520, 1989.