Birgit Janko, BSc

Factorization of non-commutative Polynomials and Testing Fullness of Matrices

MASTER THESIS

written to obtain the academic degree of

Diplom-Ingenieurin

Masterstudium Mathematische Computerwissenschaften

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr. Franz Lehner

Institute of Discrete Mathematics

Graz, February 2018

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitely marked all material which has been quotes either literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present thesis.

Date	Signature

Contents

Introduction

Factorization of commutative polynomials is a rather uniform theory, whereas treating non-commutative polynomials is not that easy. Here we use the simplest non-commutative case, factorization in the free associative algebra, which is a similarity unique factorization domain, i.e., each non-zero element has a unique factorization (up to similarity and order). A discussion of other types of non-commutative factorization can be found in [Sme15]. Recently the factorization of non-commutative polynomials became more interesting. Non-commutative polynomials are used for example in cryptanalysis or in system engineering. An application of non-commutative polynomial factorization to cryptanalysis can be found in [CCT08], in which an attack on the non-commutative Polly-Crackers is constructed. In [ARJ15] some special cases of non-commutative polynomial factorization are treated, for example the variable-disjoint factorization. The variable-disjoint factorization is even unique and can be computed in polynomial time. An algorithm for the factorization of non-commutative polynomials is described in [Car10], which uses the relations between the homogeneous parts.

In this thesis we start with a non-commutative polynomial, represented by an admissible linear system $As = v$ with a linear full square matrix A, i.e., the matrix entries are linear and the matrix does not have a factorization into rectangular matrices of smaller dimension. We try to find transformation matrices P and Q over the commutative ground field $\mathbb{K}$, such that PAQ has an upper right block of zeros, corresponding to a factorization of the represented non-commutative polynomial. We extract the coefficients of the entries corresponding to the upper right zero block and get *commutative* polynomials. Therefore we change the non-commutative problem to an equation system with *commuting* variables which we solve with Gröbner bases.

For the factorization of non-commutative polynomials, represented by an admissible linear system, we need full matrices. Let L be a linear square matrix whose matrix entries are *non-commutative* polynomials. Similarly to the factorization of non-commutative polynomials, we use in the test of fullness (of a matrix L) matrices P and Q to generate upper right zero blocks in the matrix PLQ. The coefficients of the entries corresponding to an upper right zero block are again *commutative* polynomials and generate together with $\det P - 1$ and $\det Q - 1$ an ideal. With the use of Gröbner basis we check whether this ideal is trivial or non-trivial and therefore whether the matrix is full or non-full. Testing the fullness of a linear $n \times n$ matrix practically due to (trivial or non-trivial) ideals is rather difficult for $n \geq 5$, since more than 50 unknowns are involved. We test different forms of matrices, also with different term and variable

orders for the Gröbner basis and compare them.

Motivation

Bruno Buchberger introduced the Gröbner basis theory for ideals in a commutative polynomial ring over a field in his dissertation 1965 (see [Buc65]). A Gröbner basis (named after his advisor Wolfgang Gröbner) of an ideal is a set of polynomials, such that every polynomial in the ideal generated by the Gröbner basis has (division) remainder zero. Therefore we can use Gröbner bases to check whether a polynomial is in the ideal or not. But there are several other applications for Gröbner bases, for example the problem of Solving Polynomial Equations or other algebraic problems related to ideals. For more information see for example [CLO15].

Buchberger also developed a terminating algorithm (Buchberger Algorithm) to find a finite Gröbner basis of an ideal, generated by a finite set of polynomials, with this finite set and a term order as input. Further details are given in Section 1.3. This algorithm is implemented in most computer algebra systems, for example Axiom, Maple, Mathematica or FriCAS. For a short introduction of Gröbner basis in some computer algebra systems see [CLO15].

Here we use Gröbner bases to describe the ideal as a minimal finite generated set and then test whether the Gröbner basis is trivial respectively non-trivial.

Outline

In this thesis I deal with the factorization of *non-commutative* polynomials and testing the fullness of matrices, i.e., checking whether matrices are invertible over the free field (the division algebra of the free associative algebra). The implementation is written for the interpreter in FriCAS, a computer algebra system which is a descendant of Axiom. For further information about FriCAS please go to the website `fricas.sourceforge.net`. The used program code is explained in detail in [Jan18] and will be illustrated by examples of non-commutative polynomials.

This thesis consists of three chapters which are based on each other. The first chapter gives an introduction to the basic definitions and properties which are used throughout the thesis. The second chapter handles the factorization of *non-commutative* polynomials. After the theory of polynomial factorization is set up, the implementation in FriCAS is explained and some examples of non-commutative polynomials illustrate the code. The third chapter considers full matrices, the main theorem for handling and testing full matrices and some examples. At the end of Chapter 2 and Chapter 3 some observations about the implementation in FriCAS are mentioned. The focus is especially on the current limits for calculations and some experiments concerning the runtime.

Notation

$X = \{x_1, \ldots, x_d\}$	finite alphabet X, in the examples usually $X = \{x, y, z\}$.
X^*	free monoid generated by X, for example $X^* = \{1, x, y, z, xx, \ldots\}$
$\mathbb{K}$	commutative field, for example $\mathbb{Q}$ or $\mathbb{R}$
$\overline{\mathbb{K}}$	algebraic closure of $\mathbb{K}$
$\mathbb{K}[X]$	set of all commutative polynomials with variables $x \in X$ and coefficients in $\mathbb{K}$
$\mathbb{K}\langle X\rangle$	free associative algebra, free $\mathbb{K}$-algebra, $\mathbb{K}$-algebra of non-commutative polynomials
$\overline{\mathbb{K}}\langle X\rangle$	free $\mathbb{K}$-algebra over the algebraic closure of the ground field $\mathbb{K}$
$\mathbb{K}(\langle X\rangle)$	free field, universal field of fractions of $\mathbb{K}\langle X\rangle$
$\mathbb{K}[a, b]$	$\mathbb{K}$-algebra in the (commutative) variables a_{ij}, b_{ij}
I_n	identity matrix of size n
$\pi_f = (u, A, v)$	linear representation of $f \in \mathbb{K}(\langle X\rangle)$, $f = uA^{-1}v$, $u \in \mathbb{K}^{1\times n}$, $v \in \mathbb{K}^{n\times 1}$, A linear and full
$V^{m\times n}$	space of all $m \times n$ matrices over V
$S \otimes T$	tensor product of S and T
$\deg(f)$	degree of a polynomial f
$\mathbb{N}$	natural numbers $\{1, 2, ...\}$
$\mathbb{N}_0$	natural numbers with 0
$\mathbb{Q}$	rational numbers
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
$\mathfrak{S}_m$	symmetric group
.	(lower dot) zeros in matrix
`DMP`	lexicographical term order (in FriCAS)
`HDMP`	reverse lexicographical term order (in FriCAS)

1 Basics

In this chapter we give some definitions and basic properties. In Section 1.2 we give a short introduction to linear representations and in Section 1.3 an introduction to Gröbner bases. Technical facts for the calculations in FriCAS and the used computer environment are described in Section 1.4.

1.1 Definitions and Fundamental Facts

Every element of the free field can be represented as an entry of the inverse of some full square linear matrix. (See [Coh95, Theorem 6.3.7].) Therefore we consider only linear matrices.

Definition 1.1.1 ([FR04])**.** Let $\mathbb{K}$ be a commutative field and $X = \{x_1, \ldots, x_d\}$ a finite alphabet. A matrix whose entries are of the form $a_0 + a_1x_1 + \cdots + a_dx_d$, where $a_i \in \mathbb{K}$ and x_i are (commutative or non-commutative) indeterminates, is called *linear matrix (pencil).*

The *free monoid* X^*, generated by X, is the set of all finite words $x_{i_1}x_{i_2}\cdots x_{i_n}$, $n \in \mathbb{N}$ with $i_k \in \{1, \ldots, d\}$. A *letter* is an element of the alphabet, a *word* is an element of X^*. The multiplication in X^* is defined as the concatenation product $(x_{i_1}x_{i_2}\ldots x_{i_m}) \cdot (x_{j_1}x_{j_2}\ldots x_{j_n}) = x_{i_1}x_{i_2}\ldots x_{i_m}x_{j_1}x_{j_2}\ldots x_{j_n}$, with the neutral element 1, the *empty word.* The *length* of a word $w = x_{i_1}x_{i_2}\ldots x_{i_n}$ is n, denoted by $|w| = n$.

Linear matrices can be considered as matrices over the rational function field $\mathbb{K}(y_1, \ldots, y_d)$ with *commuting* variables $y_1, \ldots, y_d$ or as matrices over the free field, the non-commutative analog of the previous one, see [Coh85] or [CR99].

Let X be a finite alphabet and $\mathbb{K}$ a commutative field. Then $\mathbb{K}\langle X\rangle$ denotes the *free* $\mathbb{K}$*-algebra* of non-commutative polynomials over $\mathbb{K}$ generated by the non-commuting variables $x \in X \cup \{1\}$. It is also called the *free associative algebra.*

Definition 1.1.2 ([FR04])**.** The rational function field in non-commuting variables $x \in X$, denoted as $\mathbb{K}(\langle X\rangle)$, is called the *free field.*

Example 1.1.3. For $X = \{x, y, z\}$ the non-commutative polynomial $p = xy - yx + 2z$ is in $\mathbb{K}\langle X\rangle$.

The free field is the unique (up to isomorphism) field generated by $\mathbb{K}\langle X\rangle$. The free field is characterized by the following property: every full square matrix M over $\mathbb{K}\langle X\rangle$ becomes invertible over the free field. (See [Coh03, Section 9.3].)

Definition 1.1.4 ([FR04])**.** A square matrix M is called *full* if there is no factorization $M = PQ$ with $P \in \mathbb{K}\langle X\rangle^{n\times p}$, $Q \in \mathbb{K}\langle X\rangle^{p\times n}$ and $p < n$.

Remark 1.1.5. A nonfull matrix cannot be invertible in any extension field of $\mathbb{K}\langle X\rangle$, so the embedding of $\mathbb{K}\langle X\rangle$ in the free field maximizes the class of invertible matrices over $\mathbb{K}\langle X\rangle$, see [CR99].

Definition 1.1.6 ([FR04])**.** The *inner rank* of a matrix $M \in \mathbb{K}\langle X\rangle^{n\times n}$ is the least $r \in \mathbb{N}$, such that M has a factorization $M = PQ$ with $P \in \mathbb{K}\langle X\rangle^{n\times r}$ and $Q \in \mathbb{K}\langle X\rangle^{r\times n}$.

One result of Cohn asserts, that the inner rank of any matrix over $\mathbb{K}\langle X\rangle$ is equal to its rank over the free field, see [Coh85, Section 5.4]. For further discussions about the inner rank see [FR04].

Definition 1.1.7 ([Coh95])**.** Two matrices A and B over $\mathbb{K}\langle X\rangle$ (of the same size) are called *associated* over a subring $R \subseteq \mathbb{K}\langle X\rangle$ if there exist invertible matrices P and Q over R such that $A = PBQ$.

Definition 1.1.8 ([CR99])**.** An $n \times n$ matrix is called *hollow* if it contains a zero submatrix of size $k \times l$ with $k + l > n$.

Example 1.1.9. The 3×3 matrix

$$\begin{pmatrix} * & * & * \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix},$$

for arbitrary non-zero entries, is a hollow matrix, since for the 2×2 zero block $2+2 = 4$ is greater than 3.

The following results provide the basis for the test of fullness in Chapter 3 (see Theorem 3.1.1).

Proposition 1.1.10 ([Coh95], Proposition 4.5.4)**.** An $n \times n$ matrix with an $r \times s$ block of zeros with $r + s > n$ cannot be full.

Lemma 1.1.11 ([Coh95], Corollary 6.3.6)**.** *A linear square matrix over $\mathbb{K}\langle X\rangle$ which is not full is associated over $\mathbb{K}$ to a linear hollow matrix.*

1.2 Linear representations

In this section we introduce linear representations. After some definitions concerning linear representations we show how to construct admissible linear systems for the rational operations and some properties of linear representations.

Definition 1.2.1 ([CR94])**.** A *linear representation* of $f \in \mathbb{K}(\langle X \rangle)$ is a triple (u, A, v) with $u \in \mathbb{K}^{1\times n}$, $A = A_0 + \sum_{i=1}^{d} A_i \otimes x_i$, $A_l \in \mathbb{K}^{n\times n}$ and $v \in \mathbb{K}^{n\times 1}$ such that A is full and $f = uA^{-1}v$. The *dimension* of the linear representation is $\dim(u, A, v) = n$.

Definition 1.2.2 ([CR94])**.** A linear representation $\pi = (u, A, v)$ of f is *minimal*, if the system matrix A has the smallest possible dimension among all linear representations of f.

Remark 1.2.3. Not even a minimal linear representation of a polynomial is unique, since rows and columns can be scaled, see Example 1.2.12.

Definition 1.2.4 ([CR99])**.** Two linear representations are called *equivalent* if they represent the same element in the free field.

If we represent the same element in the free field with two different linear representations, the representations are trivially equivalent. If $\pi' = (u', A', v')$ and $\pi'' = (u'', A'', v'')$ are both minimal and equivalent, then there exist invertible matrices $W, U \in \mathbb{K}^{n\times n}$, such that $u'' = u'U$, $A'' = WA'U$, $v'' = Wv'$. (See [CR99, Section 1].) Indeed they represent the same element in the free field, since:

$$u''(A'')^{-1}v'' = u'UU^{-1}(A')^{-1}W^{-1}Wv' = u'(A')^{-1}v'.$$

Example 1.2.5. The linear representations of $x(1-yx)$ and $(1-xy)x$, which are constructed according to the algorithm of Proposition 1.2.18 as the product of x and $(1-yx)$ respectively the product of $(1-xy)$ and x, are equivalent, since they represent both the element $x - xyx$ in the free field. For details see Example 1.2.21.

Definition 1.2.6 ([CR94])**.** Let π be a minimal representation of $f \in \mathbb{K}(\langle X \rangle)$. Then the *rank* of f is defined as $\operatorname{rank} f = \dim \pi$.

Definition 1.2.7 ([CR94])**.** Let $\pi = (u, A, v)$ be a linear representation of an element f in $\mathbb{K}(\langle X \rangle)$ of dimension n. The *left family* is the n-tuple $s = (s_1, \ldots, s_n) \subseteq \mathbb{K}(\langle X \rangle)^n$ with $s_i = (A^{-1}v)_i$. The *right family* is the n-tuple $t = (t_1, \ldots, t_n) \subseteq \mathbb{K}(\langle X \rangle)^n$ with $t_j = (uA^{-1})_j$.

Remark 1.2.8. The n-tuple s, respectively t, from Definition 1.2.7 and the solution vector s, from $As = v$, respectively the solution vector t, from $tA = u$, are used synonymously.

Proposition 1.2.9 ([CR94], Proposition 4.7)**.** A representation $\pi = (u, A, v)$ of an element $f \in \mathbb{K}\langle\langle X \rangle\rangle$ is minimal if and only if both, the left family and the right family are $\mathbb{K}$-linearly independent.

Remark 1.2.10. Notice, that in a minimal linear representation the left family and the right family have to be $\mathbb{K}$-linearly independent among themselves, but they are never $\mathbb{K}$-linearly independent from each other. At least the first component of the left family and the last component of the right family are $\mathbb{K}$-linearly dependent.

Definition 1.2.11 ([Coh72])**.** A linear representation $\pi = (u, A, v)$ of an element $f \in \mathbb{K}\langle\langle X \rangle\rangle$ is called an *admissible linear system* (for f, denoted $f \sim \pi$), denoted by $As = v$, if $u = e_1 = (1, 0, \ldots, 0)$. Then f is the first component of the (unique) solution vector s.

Example 1.2.12. Let $f = 1 - xy$ be our polynomial to examine minimal admissible linear systems of f. Then the following four admissible linear systems are examples of members of the same equivalence class.

$$\begin{pmatrix} 1 & -x & 1 \\ . & 1 & -y \\ . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ -1 \end{pmatrix}, \quad s = \begin{pmatrix} 1-xy \\ -y \\ -1 \end{pmatrix} \tag{1.2.13}$$

$$\begin{pmatrix} 1 & x & -1 \\ . & 1 & -y \\ . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} 1-xy \\ y \\ 1 \end{pmatrix} \tag{1.2.14}$$

$$\begin{pmatrix} 1 & -x & -1 \\ . & 1 & y \\ . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} 1-xy \\ -y \\ 1 \end{pmatrix} \tag{1.2.15}$$

$$\begin{pmatrix} 1 & -3x & -1 \\ . & 1 & \frac{1}{3}y \\ . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} 1-xy \\ -\frac{1}{3}y \\ 1 \end{pmatrix} \tag{1.2.16}$$

There exist transformation matrices W and U to transform one system matrix into the other. The transformation of the system matrix from (1.2.13) into that from (1.2.14) is

$$W = \begin{pmatrix} 1 & 0 & 0 \\ . & -1 & 0 \\ . & . & -1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 0 & 0 \\ . & -1 & 0 \\ . & . & -1 \end{pmatrix},$$

since the second and third row and the second and third column are multiplied by (-1). The transformation matrices for the system matrix from (1.2.14) to that from (1.2.15) are

$$W = \begin{pmatrix} 1 & 0 & 0 \\ . & -1 & 0 \\ . & . & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 0 & 0 \\ . & -1 & 0 \\ . & . & 1 \end{pmatrix},$$

since we multiply the second row and the second column by (-1). For the transformation from (1.2.15) to (1.2.16) we need

$$W = \begin{pmatrix} 1 & 0 & 0 \\ . & \frac{1}{3} & 0 \\ . & . & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 0 & 0 \\ . & 3 & 0 \\ . & . & 1 \end{pmatrix},$$

since we multiply the second row by $\frac{1}{3}$ and the second column by 3. To transform the system matrix from (1.2.16) to the system matrix from (1.2.13) we use the transformation matrices

$$W = \begin{pmatrix} 1 & 0 & 0 \\ . & 3 & 0 \\ . & . & -1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 0 & 0 \\ . & \frac{1}{3} & 0 \\ . & . & -1 \end{pmatrix}.$$

Definition 1.2.17 (Admissible Transformations, [Sch17a])**.** Let $\pi = (u, A, v)$ be a linear representation of dimension n of $f \in \mathbb{K}\langle\langle X \rangle\rangle$ and $T, U \in \mathbb{K}^{n\times n}$ invertible matrices. The transformed representation $T\pi U = (uU, TAU, Tv)$ is again a linear representation (of f). If π is an admissible linear system, the transformation (T, U) is called *admissible* if the first row of U is $e_1 = (1, 0, \ldots, 0)$.

Admissible linear systems can be constructed as follows.

Proposition 1.2.18 ([CR99], Section 1)**.** Let $f, g \in \mathbb{K}\langle\langle X \rangle\rangle$, $g \neq 0$, be the elements of the free field given by the admissible linear systems $A_f s_f = v_f$ and $A_g s_g = v_g$ and let $\lambda \in \mathbb{K}$. Then admissible linear systems for the rational operations are constructed (denoted as $\sim$) by:
The *scalar multiplication* λf is represented by

$$\lambda f \quad \sim \quad A_f s_{\lambda f} = \lambda v_f,$$

with solution vector $s_{\lambda f} = \lambda s_f$.
The *sum* $f + g$ is represented by

$$f + g \quad \sim \quad \begin{pmatrix} A_f & -A_f u_f^\top u_g \\ . & A_g \end{pmatrix} s_{f+g} = \begin{pmatrix} v_f \\ v_g \end{pmatrix}, \tag{1.2.19}$$

with solution vector $s_{f+g} = \begin{pmatrix} s_f + u_f^\top g \\ s_g \end{pmatrix}$.
The *product* fg is represented by

$$fg \quad \sim \quad \begin{pmatrix} A_f & -v_f u_g \\ . & A_g \end{pmatrix} s_{fg} = \begin{pmatrix} . \\ v_g \end{pmatrix}, \tag{1.2.20}$$

with solution vector $s_{fg} = \begin{pmatrix} s_f g \\ s_g \end{pmatrix}$.
The *inverse* g^{-1} is represented by

$$g^{-1} \quad \sim \quad \begin{pmatrix} -v_g & A_g \\ . & u_g \end{pmatrix} s_{g^{-1}} = \begin{pmatrix} . \\ 1 \end{pmatrix},$$

with solution vector $s_{g^{-1}} = \begin{pmatrix} g^{-1} \\ s_g g^{-1} \end{pmatrix}$.
The elements 0 and $k \in \mathbb{K}$, $k \neq 0$ are represented by

$$0 = (\,,\,,\,) \quad \text{and} \quad k = (1,\, 1,\, k)$$

of rank 0 and 1 respectively.

Example 1.2.21. Let us construct an admissible linear system for $p = (1 - xy)x$ using the product rule (1.2.20). Minimal admissible linear systems for the factors x and $1 - xy$ are given by

$$x \quad \sim \quad \begin{pmatrix} 1 & -x \\ . & 1 \end{pmatrix} s_x = \begin{pmatrix} . \\ 1 \end{pmatrix}, \quad s_x = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

and

$$1 - xy \quad \sim \quad \begin{pmatrix} 1 & -x & -1 \\ . & 1 & y \\ . & . & 1 \end{pmatrix} s_{1-xy} = \begin{pmatrix} . \\ . \\ 1 \end{pmatrix}, \quad s_{1-xy} = \begin{pmatrix} 1 - xy \\ -y \\ 1 \end{pmatrix}.$$

For the product of those two factors we get (from Proposition 1.2.18) the admissible linear system

$$\left(\begin{array}{ccc|cc} 1 & -x & -1 & . & . \\ . & 1 & y & . & . \\ . & . & 1 & -1 & . \\ \hline . & . & . & 1 & -x \\ . & . & . & . & 1 \end{array}\right) s = \begin{pmatrix} . \\ . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1 - xy)x \\ -yx \\ x \\ x \\ 1 \end{pmatrix}.$$

Observe that this admissible linear system is not minimal yet, since the left family is $\mathbb{K}$-linearly dependent. To obtain a minimal system, we proceed as follows: First, we add column 3 to column 4 with the transformation matrices

$$\tilde{W} = I_5 = \begin{pmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & . & 1 & . \\ . & . & . & . & 1 \end{pmatrix} \quad \text{and} \quad \tilde{U} = \begin{pmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & 1 & . \\ . & . & . & 1 & . \\ . & . & . & . & 1 \end{pmatrix}$$

to get

$$\begin{pmatrix} 1 & -x & -1 & -1 & . \\ . & 1 & y & y & . \\ . & . & 1 & 0 & . \\ . & . & . & 1 & -x \\ . & . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1-xy)x \\ -yx \\ 0 \\ x \\ 1 \end{pmatrix}.$$

Then deleting row 3 and column 3 (since the corresponding entry s_3 is zero) in the system matrix yields the system

$$\begin{pmatrix} 1 & -x & -1 & . \\ . & 1 & y & . \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1-xy)x \\ -yx \\ x \\ 1 \end{pmatrix}.$$

Since the left family s and the right family $t = (1,\, x,\, 1-xy,\, x-xyx)$ are both $\mathbb{K}$-linearly independent, a minimal admissible linear system for $p = (1-xy)x$ is given by

$$\begin{pmatrix} 1 & -x & -1 & . \\ . & 1 & y & . \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1-xy)x \\ -yx \\ x \\ 1 \end{pmatrix}. \tag{1.2.22}$$

Recall that p is the first component of the solution vector s.

Similarly we get a minimal admissible linear system for $q = x(1-yx)$,

$$\begin{pmatrix} 1 & -x & . & . \\ . & 1 & y & -1 \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} x(1-yx) \\ 1-yx \\ x \\ 1 \end{pmatrix}. \tag{1.2.23}$$

With the two invertible matrices

$$W = \begin{pmatrix} 1 & . & 1 & . \\ . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & . & . & . \\ . & 1 & . & -1 \\ . & . & 1 & . \\ . & . & . & 1 \end{pmatrix}$$

one can easily verify that these two linear representations are equivalent.

Remark 1.2.24. In Example 1.2.21 the matrices W and U represent the row and column operations respectively, transforming in this case the admissible linear system of $p = (1-xy)x$ to $q = x(1-yx)$, constructed with the product rule (1.2.20). In general the representations are totally arbitrary. We get the system matrix in (1.2.23), if we add the second column, after multiplying it with (-1), to the fourth column and add the third row to the first row in (1.2.22).

Remark 1.2.25. Notice that there always exist two invertible transformation matrices for all pairs of minimal admissible linear systems of a polynomial. (See [CR99, Section 1].) If we use in Example 1.2.21 the same admissible linear system for $1-yx$ as for $1-xy$ only exchanging x with y and y with x in the system matrix, i.e.

$$(1-xy)x \quad \sim \begin{pmatrix} 1 & -x & -1 & . \\ . & 1 & y & . \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1-xy)x \\ -yx \\ x \\ 1 \end{pmatrix} \text{ and}$$

$$x(1-yx) \quad \sim \begin{pmatrix} 1 & -x & . & . \\ . & 1 & -y & -1 \\ . & . & 1 & x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} (1-xy)x \\ 1-yx \\ -x \\ 1 \end{pmatrix},$$

then we have the two transformation matrices

$$W = \begin{pmatrix} 1 & 0 & 1 & 0 \\ . & 1 & 0 & 0 \\ . & . & -1 & 0 \\ . & . & . & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ . & 1 & 0 & -1 \\ . & . & -1 & 0 \\ . & . & . & 1 \end{pmatrix}.$$

However we cannot assume that the transformation matrices always have the following form

$$W = \begin{pmatrix} 1 & a_{1,2} & \dots & a_{1,n-1} & 0 \\ . & a_{2,2} & \dots & a_{2,n-1} & 0 \\ . & . & \ddots & \vdots & \vdots \\ . & . & . & a_{n-1,n-1} & 0 \\ . & . & \dots & . & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 1 & b_{1,2} & \dots & b_{1,n-1} & 0 \\ . & b_{2,2} & \dots & b_{2,n-1} & 0 \\ . & . & \ddots & \vdots & \vdots \\ . & . & . & b_{n-1,n-1} & 0 \\ . & . & \dots & . & 1 \end{pmatrix}$$

with $a_{i,i}b_{i,i} = 1 \quad \forall\, i \in \{2, \dots, n-1\}$. A counter-example is the anticommutator $xy + yx$ whose transformation matrices contain permutations which are not upper triangle matrices. One minimal admissible linear system of the anticommutator is

$$\begin{pmatrix} 1 & -x & -y & . \\ . & 1 & . & -y \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} xy+yx \\ y \\ x \\ 1 \end{pmatrix}, \tag{1.2.26}$$

another minimal admissible linear system is

$$\begin{pmatrix} 1 & -y & -x & . \\ . & 1 & . & -x \\ . & . & 1 & -y \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} xy+yx \\ x \\ y \\ 1 \end{pmatrix}. \tag{1.2.27}$$

We get the system matrix in (1.2.27), if we exchange the second row with the third row and the second column with the third column in (1.2.26). These operations are done by the transformation matrices

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ . & 0 & 1 & 0 \\ . & 1 & 0 & 0 \\ . & . & . & 1 \end{pmatrix} \text{ and } U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ . & 0 & 1 & 0 \\ . & 1 & 0 & 0 \\ . & . & . & 1 \end{pmatrix},$$

which are not upper triangular matrices any more.

In this thesis only minimal admissible linear systems are considered to simplify the manipulations. The construction of minimal linear representations is not discussed here and referred to the illustration in [Sch17b, Section 4] and the overview in [Sch17a, Section 3].

Factorization of commutative polynomials is a rather uniform theory, whereas the non-commutative case is not that easy. In this thesis we use the simplest non-commutative case, factoring in the free associative algebra, which is a (similarity unique) factorization domain. For further information of non-commutative factorization, see [Sme15].

Definition 1.2.28 ([Coh85], Section 3.2)**.** Let R be a ring and $\mathfrak{a}$, $\mathfrak{b} \subseteq R$ two ideals. The two ideals are called *similar*, denoted by $\mathfrak{a} \sim \mathfrak{b}$, if $R/\mathfrak{a} \cong R/\mathfrak{b}$ as right R-modules. Two elements p, $q \in R$ are called *similar*, if the right ideals they generate pR and qR are similar, this means $pR \sim qR$.

Definition 1.2.29 ([BS15], Section 2)**.** Let R be a domain and $H = R\backslash\{0\}$.

(i) An element p *left divides* q, denoted by $p \mid_l q$, if $q \in pH = \{ph \mid h \in H\}$. Two elements p, q are called *left coprime*, if for every h, such that $h \mid_l p$ and $h \mid_l q$, h is invertible, this means that h is an element of the *group of units*.

(ii) An element p *right divides* q, denoted by $p \mid_r q$, if $q \in Hp = \{hp \mid h \in H\}$. Two elements p, q are called *right coprime*, if for every h, such that $h \mid_r p$ and $h \mid_r q$, h is invertible, this means that h is an element of the *group of units*.

Definition 1.2.30 ([BS15], Section 2)**.** Let R be a domain and $H = R\backslash\{0\}$. An element $p \in H\backslash H^\times$, i.e., a non-zero non-unit (in R), is called an *atom* or *irreducible*, if $p = q_1 q_2$ with q_1, $q_2 \in H$ implies that either $q_1 \in H^\times$ or $q_2 \in H^\times$. The (cancellative) monoid H is called *atomic*, if every non-unit can be written as a finite product of atoms of H. The domain R is called *atomic*, if the monoid $R\backslash\{0\}$ is atomic.

Example 1.2.31. The atoms in the free monoid X^* are the letters $x \in X$.

Definition 1.2.32 ([Sme15], Definition 4.1)**.** A domain R is called *similarity Unique Factorization Domain*, if R is atomic and it satisfies the property that if $p_1p_2\cdots p_m = q_1q_2\cdots q_n$ for atoms p_i, $q_j \in R$, then $m = n$ and there exists a permutation $\sigma \in \mathfrak{S}_m$ such that p_i is similar to $q_{\sigma(i)}$ for all $i \in \{1, 2, \ldots, m\}$.

Proposition 1.2.33 ([Coh63], Theorem 6.3)**.** The free associative algebra $\mathbb{K}\langle X\rangle$ is a similarity Unique Factorization Domain.

Definition 1.2.34 ([CR99])**.** A *linearization* of $f \in \mathbb{K}(\langle X\rangle)$ is a matrix $L = L_0 + \sum_{i=1}^{d} L_i \otimes x_i$, with $L_i \in \mathbb{K}^{m\times m}$, of the form

$$L = \begin{pmatrix} c & u \\ v & A \end{pmatrix} \in \mathbb{K}\langle X\rangle^{m\times m}$$

such that A is invertible over the free field and $f = c - uA^{-1}v$. Recall, that A is a full matrix.
If $c = 0$ then L is called a *pure linearization*.
The *size* of the linearization is $\operatorname{size} L = m$, the *dimension* is $\dim L = m - 1$.

Remark 1.2.35. Given a linear representation $(u,\ A,\ v)$ of $f \in \mathbb{K}(\langle X\rangle)$, $L = \begin{pmatrix} \cdot & u \\ -v & A \end{pmatrix}$ is a pure linearization of f.

Corollary 1.2.36 ([CR99], Corollary 1.3)**.** *The elements of the free field $\mathbb{K}(\langle X\rangle)$ are given by equivalence classes of pure and linear representations, with the field operations given by the rational operations in Proposition 1.2.18.*

Theorem 1.2.37 ([CR99], Theorem 1.4)**.** *If $\pi' = (u',\ A',\ v')$ and $\pi'' = (u'',\ A'',\ v'')$ are equivalent (pure) linear representations, of which the first is minimal, then the second is isomorphic to a representation $(u,\ A,\ v)$ which has the block decomposition*

$$u = \begin{pmatrix} * & u' & \cdot \end{pmatrix}, \quad A = \begin{pmatrix} * & \cdot & \cdot \\ * & A' & \cdot \\ * & * & * \end{pmatrix} \text{ and } v = \begin{pmatrix} \cdot \\ v' \\ * \end{pmatrix}.$$

1.3 Gröbner bases

In this section we introduce Gröbner bases and discuss their basic properties. This is based on the lecture notes [Gil15]. Here Gröbner bases are used to find a factorization of a non-commutative polynomial (as an element in the free field $\mathbb{K}(\langle X\rangle)$). For more detailed information about Gröbner bases and their alternative uses see [CLO15].

Notice that in this section we consider *commutative* polynomials.

Before starting with Gröbner bases we need to define some basics like term orders.

Definition 1.3.1. Let $X = \{\xi_1, \ldots, \xi_d\}$ be a finite alphabet. Then $T = T(X) = \{\xi_1^{e_1} \cdot \ldots \cdot \xi_d^{e_d} \mid e_1, \ldots, e_d \in \mathbb{N}_0\}$ is the set of *terms*.
The *product of terms* is defined as $\left(\xi_1^{e_1} \cdot \ldots \cdot \xi_d^{e_d}\right) \cdot \left(\xi_1^{f_1} \cdot \ldots \cdot \xi_d^{f_d}\right) = \xi_1^{e_1+f_1} \cdot \ldots \cdot \xi_d^{e_d+f_d}$.
The set of *monomials* is $M = M(X) = \{a_t \cdot t \mid a_t \in \mathbb{K}\backslash\{0\},\, t \in T\}$.
The *product of monomials* is defined as $(a_1 t_1) \cdot (a_2 t_2) = (a_1 a_2)(t_1 \cdot t_2)$ and the *degree of a monomial* is $\deg(a\xi_1^{e_1} \cdot \ldots \cdot \xi_d^{e_d}) = e_1 + \cdots + e_d$.
The polynomials over $\mathbb{K}$ in X are

$$\mathbb{K}[X] = \left\{ \sum_{t \in S} a_t \cdot t \mid a_t \in \mathbb{K}\backslash\{0\},\, S \subset T,\, S \text{ finite} \right\} \cup \{0\}.$$

The *degree of a polynomial* is the maximum degree of its monomials.

Definition 1.3.2. Let T be the set of terms in $\mathbb{K}[X]$. A *term order* is a total order $\leq$ on T such that

(i) $1 \leq t \quad \forall\, t \in T$

(ii) If $s \leq t \Rightarrow s \cdot t' \leq t \cdot t' \quad \forall\, s, t, t' \in T$.

There are several term orders, which result in small differences in the ordering of terms, but give rise to dramatically different Gröbner bases.

Definition 1.3.3. Let $s = \xi_1^{e_1} \cdot \ldots \cdot \xi_d^{e_d}$ and $t = \xi_1^{f_1} \cdot \ldots \cdot \xi_d^{f_d}$ be two terms.
The *lexicographic term order* is the term order $<_{\text{lex}}$ if

$$s <_{\text{lex}} t \quad \Longleftrightarrow \quad \exists\, 1 \leq m \leq d \text{ such that } \forall\, i < m : e_i = f_i \text{ and } e_m < f_m.$$

The *inverse lexicographic term order* is the term order $<_{\text{invlex}}$ if

$$s <_{\text{invlex}} t \quad \Longleftrightarrow \quad \exists\, 1 \leq m \leq d \text{ such that } \forall\, i > m : e_i = f_i \text{ and } e_m < f_m.$$

The *graded lexicographic term order* is the term order $<_{\text{gradlex}}$ if

$$s <_{\text{gradlex}} t \quad \Longleftrightarrow \quad \big[\deg(s) < \deg(t)\big] \text{ or } \big[\deg(s) = \deg(t) \text{ and } s <_{\text{lex}} t\big].$$

The *graded reverse lexicographic term order* is the term order $<_{\text{gradlex}}$ if

$$s <_{\text{gradrevlex}} t \quad \Longleftrightarrow \quad \big[\deg(s) < \deg(t)\big] \text{ or } \big[\deg(s) = \deg(t) \text{ and } t <_{\text{lex}} s\big].$$

Example 1.3.4. Let $X = \{\xi_1, \xi_2, \xi_3\}$. Then $\xi_1^2\xi_2\xi_3^3 <_{\text{lex}} \xi_1^2\xi_2^4\xi_3$ and $\xi_1^2\xi_2\xi_3^3 <_{\text{gradlex}} \xi_1^2\xi_2^4\xi_3$, but $\xi_1^2\xi_2\xi_3^3 >_{\text{invlex}} \xi_1^2\xi_2^4\xi_3$.

Definition 1.3.5. Let $<$ be a term order on T, $0 \neq f = \sum_{t \in S} a_t \cdot t \in \mathbb{K}[X]$ with $a_t \in \mathbb{K}\backslash\{0\}$ and S a finite subset of T. Then the *set of terms in* f, denoted by $T(f)$, is S. The *leading term of* f is the maximal term in S with respect to the term order $<$, thus $\text{lt}(f) = \max_<(S)$. The *leading monomial* of f, denoted by $\text{lm}(f)$, is $\text{lm}(f) = a_t \cdot t$ with $t = \text{lt}(f)$.

Now we can define Gröbner bases.

Definition 1.3.6. Let $\mathbb{K}$ be a commutative field, I an ideal in $\mathbb{K}[X]$, $<$ the term order on the set of terms and $G \subseteq I$ finite. Then G is a *Gröbner basis* of I in $\mathbb{K}[X]$ with respect to the term order $<$, if

$$\mathrm{lt}(I) = \mathrm{Mult}(\mathrm{lt}(G)) \quad \text{where}$$

$\mathrm{lt}(M) = \{\mathrm{lt}(f) \mid f \in M\}$, for $M \subseteq \mathbb{K}[X]$ and
$\mathrm{Mult}(S) = \{t \cdot s \mid t \in T,\, s \in S\}$, for $S \subseteq \mathbb{K}[X]$.

Corollary 1.3.7. *Let G be the Gröbner basis of an ideal $I \subset \mathbb{K}[X]$. Then G generates I, i.e., $I = \mathrm{Id}(G)$.*

Proof. The ideal generated by G is trivially a subset of I.

Now let $\mathrm{Id}(G)$ be the ideal generated by G and assume that $I \setminus \mathrm{Id}(G)$ is not empty. Choose $f \in I \setminus \mathrm{Id}(G)$ with minimal leading term with respect to $<$. According to the assumption there exists $g \in G$, whose leading term divides the leading term of f. Let $\mathrm{lm}(f) = a \cdot s$ and $\mathrm{lm}(g) = b \cdot t$ with $a, b \in \mathbb{K}$, $s, t \in T$. It follows that t divides s, say $t \cdot u = s$ with $u \in T$.

Let $h := f - (ab^{-1}u)g \neq 0$. (Otherwise $f = ab^{-1}ug$ and therefore f is an element of the ideal generated by G.) So it holds that $\mathrm{lm}(h) < \mathrm{lm}(f)$, $h \in I$. Since the leading term in f is minimal in $I \setminus \mathrm{Id}(G)$, h is in the ideal generated by G. Thus it follows that $f = h + (ab^{-1}u)g$ is an element of $\mathrm{Id}(G)$, which is a contradiction to our assumption. Therefore $I \setminus \mathrm{Id}(G)$ is empty and I is the ideal generated by G. □

Theorem 1.3.8. *For every ideal $I \subset \mathbb{K}[X]$ and for every term order $<$ on T there exists a Gröbner basis of I with respect to $<$.*

For the proof of this theorem we use the fact that $(T, |)$ has the Dickson-property.

Definition 1.3.9. Let $(M, \leq)$ be a partial order, that is, $\leq$ is reflexive, transitive and antisymmetric. Then $(M, \leq)$ has the *Dickson-property*, if every non-empty subset N of M has a finite base. $B \subseteq N$, is a *base* of N, if for every $a \in N$ there exists a $b \in B$ with $b \leq a$.

Proof of Theorem 1.3.8. Let $I \neq \emptyset$. Then $\mathrm{lt}(I) \neq \emptyset$. Let $|$ represent the following property: $\xi_1^{e_1} \ldots \xi_d^{e_d} \mid \xi_1^{f_1} \ldots \xi_d^{f_d} \quad \Leftrightarrow \quad e_i \leq f_i \quad \forall\, i \in \{1, \ldots, d\}$. Let B be the finite Dickson-base of $\mathrm{lt}(I)$ with respect to $(T, |)$, like $B = \{t_1, \ldots, t_n\} \subseteq \mathrm{lt}(I)$. Choose $g_i \in I$ with $lt(g_i) = t_i$ and we claim that $G = \{g_1, \ldots, g_n\}$ is a Gröbner basis of I with respect to $<$. For proving this claim, let $0 \neq f \in I$. Then there exists $t_i \in B$ such that t_i divides the leading term of f. But this means that the leading term of f can be written as $\mathrm{lt}(f) = u \cdot \mathrm{lt}(g_i)$ with $u \in T$. So the leading term of f is a multiple of the leading terms of the Gröbner basis. Therefore we have shown the statement of the Theorem, since the other direction follows immediately from the fact $G \subset I$. □

We can construct a Gröbner basis from a given finite polynomial system with the Buchberger Algorithm. The algorithm is based on the subtraction polynomial and is shown in Algorithm 1. Before concentrating on the algorithm we need some more definitions.

Definition 1.3.10. Let $f, g, p \in \mathbb{K}[X]$, $p \neq 0$, $P \subseteq \mathbb{K}[X]$, $<$ be a term order and $t \in T$.

(i) f *reduces itself to* g under elimination of t with respect to P, denoted by $f \underset{P}{\longrightarrow} g[t]$, if $\exists\, a \in \mathbb{K}$ with $a \cdot t \in M(f)$, $\mathrm{lm}(p) = b \cdot s$ with $b \in \mathbb{K}$, $s \in T$, $t = u \cdot s$ for $u \in T$ and $g := f - \frac{a}{b} up$.

(ii) $f \underset{p}{\longrightarrow} g$ if there is a $t \in T(f)$ with $f \underset{p}{\longrightarrow} g[t]$.

(iii) $f \underset{P}{\longrightarrow} g$ if there exists a $p \in P$ with $f \underset{p}{\longrightarrow} g$.

(iv) f is *reducible* mod p (mod P), if there is a $g \in \mathbb{K}[X]$ with $f \underset{p}{\longrightarrow} g$ (respectively $f \underset{P}{\longrightarrow} g$).

(v) f is in *normal form* mod p (mod P), if f is not reducible mod p (mod P).

(vi) $f \xrightarrow[P]{k} g$ if there exist $f_0, \ldots, f_k$ with $f = f_0 \underset{P}{\longrightarrow} f_1 \underset{P}{\longrightarrow} \ldots \underset{P}{\longrightarrow} f_k = g$.

(vii) $f \xrightarrow[P]{*} g$ if there exists a $k \in \mathbb{N}_0$ such that $f \xrightarrow[P]{k} g$.

Example 1.3.11. Let $f = 5\xi_1^2\xi_2^7 - \xi_1^3\xi_2^4 + \xi_1^2 - \xi_2 + 3$, $p = 2\xi_1^2\xi_2 - 1$ and $<$ the graded lexicographical term order. If we consider $t = \xi_1^2\xi_2^7$, then

$$f \underset{p}{\longrightarrow} f - \frac{5}{2}\xi_2^6 p = -\xi_1^3\xi_2^4 + \xi_1^2 - \xi_2 + 3 + \frac{5}{2}\xi_2^6 =: g.$$

Definition 1.3.12. Let $s = \xi_1^{e_1} \cdot \ldots \cdot \xi_d^{e_d}$ and $t = \xi_1^{f_1} \cdot \ldots \cdot \xi_d^{f_d}$ be two terms. Then the *least common multiple* of s and t, denoted by $\mathrm{lcm}(s, t)$, is

$$\mathrm{lcm}(s, t) = \xi_1^{\max\{e_1, f_1\}} \cdot \ldots \cdot \xi_d^{\max\{e_d, f_d\}}.$$

This means that s and t divides $\mathrm{lcm}(s, t)$ and if $t' \in T$ with s and t divides t', then $\mathrm{lcm}(s, t)$ divides t'.

Definition 1.3.13. Let $0 \neq f, g \in \mathbb{K}[X]$, $t' = \mathrm{lcm}(s, t) = u \cdot s = v \cdot t$ with $u, v \in T$. Then the *subtraction polynomial* of f and g is defined as $\mathrm{Spol}(f, g) = buf - avg$.

Theorem 1.3.14. *Let $G \subseteq \mathbb{K}[X]$ be finite, $<$ term order and $0 \notin G$. Then*

$$G \text{ is Gröbner basis with respect to } < \iff \forall\, f, g \in G \text{ with } f \neq g : \mathrm{Spol}(f, g) \xrightarrow[G]{*} 0.$$

The proof of Theorem 1.3.14 can be found in [CLO15]. This theorem provides the main idea of the Buchberger Algorithm to find a Gröbner basis from a finite set of polynomials.

Algorithm 1 Buchberger Algorithm

Input: $F \subseteq \mathbb{K}[X]$ finite, $<$ term order, $0 \notin F$
Output: $G \subseteq \mathrm{Id}(F)$ finite, s.t. G is Gröbner basis of $\mathrm{Id}(F)$ with respect to $<$ and $F \subseteq G$

$G := F$
$B := \big\{\{g_1, g_2\} \,|\, g_i \in G,\, g_1 \neq g_2\big\}$
while $B \neq \emptyset$ **do**
  choose $\{g_1, g_2\} \in B$
  $B := B \backslash \big\{\{g_1, g_2\}\big\}$
  $h := \mathrm{Spol}(g_1, g_2)$
  $h_0 :=$ normal form of h with respect to $\xrightarrow[G]{}$
  if $h_0 \neq 0$ **then**
    $B := B \cup \big\{\{g, h_0\} \,|\, g \in G\big\}$
    $G := G \cup \{h_0\}$
  end if
end while

Proof. For the termination of this algorithm we assume that there exists a finite subset $F \subseteq \mathbb{K}\langle X\rangle$, such that the while-loop does not terminate. Then there are infinitely many instances of $h_0 \neq 0$. Let $h_0, h_1, h_2, \ldots$ be these instances of h_0 and $G_0, G_1, G_2, \ldots$ be the instances of G after the corresponding while-loop. Therefore $h_i \in G_i$ and even $\{h_0, \ldots, h_i\} \subseteq G_i$ holds. According to the construction h_i is in normal form mod $G_{i-1} \subseteq G_i$. In particular, $t_i := \mathrm{lt}(h_i)$ is in normal form mod G_{i-1}. So it follows that $\forall\, s \in \mathrm{lt}(G_{i-1})$ s does not divide t_i (otherwise t_i could be reduced). Let B' a finite Dickson basis of $\{t_i \,|\, i = 0, 1, 2, \ldots\}$ in $(T, |)$. Choose $n \in \mathbb{N}$ such that $B' \subseteq \{t_0, t_1, \ldots, t_n\}$. Now we can find for t_{n+1} some $t_i \in B'$ such that t_i divides t_{n+1} for i smaller or equal n. So $t_i \in \mathrm{lt}(G_i)$ and t_{n+1} is in normal form mod G_n, but $t_i \in \mathrm{lt}(G_i) \subseteq \mathrm{lt}(G_n)$. This results in a contradiction, since t_i divides t_{n+1} and $t_i \in \mathrm{lt}(G_n)$ follow to the reducibility of t_{n+1} mod G_n.

Concerning the correctness of the algorithm, the following hold before the first loop run and after each loop run :

(i) $F \subseteq G$ finite, $0 \notin G$, $\mathrm{Id}(G) = \mathrm{Id}(F)$

(ii) $\forall\, g_1, g_2 \in G$ with $g_1 \neq g_2$ and $\{g_1, g_2\} \notin B$: $\quad \mathrm{Spol}(g_1, g_2) \xrightarrow[G]{*} 0$

If the algorithm terminates, B is an empty set and therefore $\forall\, g_1, g_2 \in G$ with $g_1 \neq g_2$

$\mathrm{Spol}(g_1, g_2) \xrightarrow[\mathrm{G}]{*} 0$, since (ii) holds. From Theorem 1.3.14 it follows that G is a Gröbner basis of $\mathrm{Id}(G) = \mathrm{Id}(F)$ with respect to $<$. □

Lemma 1.3.15 (Buchberger's Criterion)**.** *Let $f, g \in \mathbb{K}[X]$ with disjoint leading terms and $<$ be the term order. Then* $\mathrm{Spol}(f, g) \xrightarrow[\{f,g\}]{*} 0$.

We already mentioned the dependence of Gröbner on the term order. In the following example we illustrate the differences between the Gröbner bases in connection with different term orders.

Example 1.3.16. Let $X = \{\xi_1, \xi_2, \xi_3\}$ and the system of polynomials $F = \{f, g, h\}$ with $f = \xi_1+1$, $g = \xi_2+1$ and $h = \xi_1\xi_2+\xi_3$. Recall, that we consider here *commutative* polynomials!

First of all, we will use the lexicographic term order $<_{\mathrm{lex}}$. Using Buchberger's Criterion we know that $\mathrm{Spol}(f, g) \xrightarrow[\{f,g\}]{*} 0$. For $\mathrm{Spol}(f, h)$ we add a new polynomial $k = -\xi_3-1$ to the current Gröbner basis. After reducing $\mathrm{Spol}(g, h)$, $\mathrm{Spol}(f, k)$, $\mathrm{Spol}(g, k)$ and $\mathrm{Spol}(h, k)$ to 0, the Buchberger Algorithm results in the Gröbner basis $G_{\mathrm{lex}} = \{\xi_1+1, \xi_2+1, \xi_1\xi_2+\xi_3, -\xi_3-1\}$.

The inverse lexicographic term order results in the Gröbner basis $G_{\mathrm{invlex}} = \{\xi_1+1, \xi_2+1, \xi_3+\xi_1\xi_2\}$, since all leading terms are pairwise disjoint (Buchberger's Criterion).

For comparison we also calculate a Gröbner basis with respect to the graded reverse lexicographic term order. Similarly to the lexicographic case, the Buchberger Algorithm results in the Gröbner basis $G_{\mathrm{gradrevlex}} = \{\xi_1+1, \xi_2+1, \xi_1\xi_2+\xi_3, -\xi_3+\xi_2\}$.

In conclusion, the Gröbner bases for $F = \{\xi_1+1, \xi_2+1, \xi_1\xi_2+\xi_3\}$ are, depending on the term order,

$$\begin{aligned} G_{\mathrm{lex}} &= \{\xi_1+1, \xi_2+1, \xi_1\xi_2+\xi_3, -\xi_3-1\} \\ G_{\mathrm{invlex}} &= \{\xi_1+1, \xi_2+1, \xi_3+\xi_1\xi_2\} \\ G_{\mathrm{gradrevlex}} &= \{\xi_1+1, \xi_2+1, \xi_1\xi_2+\xi_3, -\xi_3+\xi_2\}. \end{aligned}$$

1.4 Technical Facts and Notation for using FriCAS

In Section 2.4 (for the factorization of non-commutative polynomials) and Section 3.4 (for testing fullness of matrices) we measure the run-time in FriCAS. Therefore we give some technical facts of FriCAS and the used computer environment in the following remarks.

Remark 1.4.1. The total run-times were measured with the implemented time function, started with `)time on` in FriCAS on a computer with 7 GB of RAM dedicated to SBCL. In the standard installation SBCL uses only 1 GB, which results in slower

run-times and more “Control stack exhausted” errors. The measured run-times are only samples, for example to show the relative differences between the different term and/or variable orders.

Remark 1.4.2. Additional information of the used computer:
CPU: Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz: 4 kernels, 6144 KB Cache
RAM: 8GB DDR3-1600 Modul

Concerning the run-time measurement the following three special cases are occurred in FriCAS and we use the following short cuts.

- *CSE* represents the system error “Control stack exhausted”, which means that there is no more space for further function calls. In FriCAS we have the output
  ```
  Control stack guard page temporarily disabled:  proceed with
  caution
  >> System error:
  Control stack exhausted (no more space for function call frames).
  This is probably due to heavily nested or infinitely recursive
  function calls, or a tail call that SBCL cannot or has not
  optimized away.
  PROCEED WITH CAUTION.
  ```
  Another reason for this error may be too large argument lists.

- *LDB* represents the error “Heap exhausted”. Additionally the time between the function call and the appearance of the error message on the screen is shown. In FriCAS we switch to the low-level debugger for the Lisp runtime environment and get the output
  ```
  [...]
  Heap exhausted, game over.
  Welcome to LDB, a low-level debugger for the Lisp runtime
  environment.
  ldb>
  ```

- $> n$ *min* indicates that the FriCAS process was terminated manually after n minutes. In this case we are not able to know the result. Either the run-time with a result is greater than or equal to the shown minutes, or after that an error (CSE or LDB) occurs.

2 Factorization of non-commutative polynomials

In this chapter we consider the factorization of non-commutative polynomials into irreducible elements. To this end a non-commutative polynomial is represented by a *minimal* admissible linear system $As = v$. For the definition and introduction of (minimal) admissible linear systems see Section 1.

With an analogue of Theorem 3.1.1 we try to find matrices P and Q, such that PAQ has an upper right block of zeros, corresponding to a factorization of the represented non-commutative polynomial. In Section 2.1 we will go into detail.

Factorizing a non-commutative polynomial of rank greater than or equal to three is very difficult, since the solution variety has non-zero dimension, i.e., there are infinitely many solutions. Here the problem is that we have too many degrees of freedom.

Remark. There are even examples, for which the solver in FriCAS is not able to find a solution although it is easily solvable by hand. For the polynomial $f = 1 - yx - xy + xy^2x$ the Gröbner bases for the 2×2 right upper block is

$$\begin{aligned}[a_{1,4}b_{4,5} + b_{3,5} - 1,\ (a_{1,3} - 1)b_{4,5} + a_{1,2}b_{3,5} - a_{1,2},\ b_{3,5}^2 - 2b_{3,5} + 1,\ a_{1,4}b_{3,5} - a_{1,4},\\ (a_{1,3} - 1)b_{3,5} - a_{1,3} + 1,\ b_{3,4} + a_{1,4},\ b_{2,5} + a_{1,4},\ b_{2,4},\ a_{2,4},\ a_{2,3} - a_{1,4},\\ a_{1,4}^2,\ (a_{1,3} - 1)a_{1,4},\ a_{1,2}a_{1,4} - a_{1,3} + 1,\ a_{1,3}^2 - 2a_{1,3} + 1].\end{aligned}$$

Consideration of algebraic varieties of positive dimension requires additional techniques and is out of scope of this thesis. Therefore we focus on determining the ranks of the factors and put the concrete solution of the transformation matrices (for the factors) last.

In Section 2.2 we comment on the FriCAS code, which is used for the examples in Section 2.3. Observations about the implementation in FriCAS concerning the run-time or upper bounds of the rank are discussed in Section 2.4.

2.1 Polynomial Factorization

In this section we give two additional definitions for the factorization and consider when a factorization of a non-commutative polynomial exists.

Definition 2.1.1. ([Sch17b]) An admissible linear system $\pi = (u, A, v)$ for a polynomial $f \in \mathbb{K}\langle X\rangle$ with dimension n is called a *pre-standard admissible linear system*, if

(i) $v = (0, \ldots, 0, \lambda)^\top$ for some $\lambda \in \mathbb{K}$ and

(ii) $a_{ii} = 1$ for $i = 1, 2, \ldots, n$ and $a_{ij} = 0$ for $i > j$.

A pre-standard admissible linear system is written as $\pi = (1, A, \lambda)$ with $1, \lambda \in \mathbb{K}$.

Definition 2.1.2. An admissible transformation (W, U) for a pre-standard admissible linear system π is called *pre-standard admissible transformation*, if the transformed system $W\pi U$ is again pre-standard.

Theorem 2.1.3 ([Sch17b])**.** *Let $f \in \mathbb{K}\langle X\rangle$ be given by the minimal pre-standard admissible linear system $\pi = (1, A, \lambda)$ of dimension $n = \operatorname{rank}(f) \geq 3$. Then f has a factorization into $f = f_1 f_2$ with $\operatorname{rank}(f_i) = n_i \geq 2$ if and only if there exists a pre-standard admissible linear transformation (P, Q) such that PAQ has an upper right block of zeros of size $(n_1 - 1) \times (n_2 - 1)$.*

For more details see [Sch17b].

Therefore, if we want to factor a polynomial f into factors $f = f_1 f_2$ with $\operatorname{rank}(f_i) = n_i \geq 2$ and $n = n_1 + n_2 - 1$, we must find transformation matrices of the form

$$(P, Q) = \left(\begin{pmatrix} 1 & a_{1,2} & \ldots & a_{1,n-1} & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 1 & a_{n-2,n-1} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & b_{2,3} & \ldots & b_{2,n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & b_{n-1,n} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \right) \tag{2.1.4}$$

with entries $a_{i,j}, b_{i,j} \in \mathbb{K}$, such that PAQ has an appropriate zero block. In practice, this means that we have to solve a system of algebraic equations.

Remark 2.1.5. The transformation matrices P and Q are upper triangular matrices with ones on the diagonals, since the system matrix of the polynomial has pre-standard form. With the ones on the diagonals we have $\det P = \det Q = 1$ and therefore these two matrices are invertible. The transformation matrix Q must not change the first component in the left family, therefore the first row of Q is the first unit (row) vector. Since the first component of the left family must not change, also the last component of the right family must not change. Therefore the last column of the transformation matrix P is the n-th unit (column) vector or maybe a scalar multiple of it.

Proposition 2.1.6 ([Sch17b])**.** Let $f \in \mathbb{K}\langle X\rangle$ be given by the minimal pre-standard admissible linear system $\pi = (1, A, \lambda)$ of dimension $n = \operatorname{rank}(f) \geq 3$ and let (P, Q) as in (2.1.4). Fix $k \in [1, n-2]$ and denote by I_k the ideal of $\mathbb{K}[a_{i,j}, b_{i,j}]$ which is generated by the coefficients of each $x \in X \cup \{1\}$ in the (i, j) entries of the matrix PAQ for $1 \leq i \leq k$ and $n-k \leq j \leq n$. Then f factors over $\overline{\mathbb{K}}\langle X\rangle$ into $f = f_1 f_2$ with $\operatorname{rank}(f_1) = k+1$ and $\operatorname{rank}(f_2) = n-k$ if and only if the ideal I_k is non-trivial.

For illustration, there exist scalar matrices P and Q such that the matrix PAQ has the form (2.1.7) if and only if f factors over $\overline{\mathbb{K}}\langle X\rangle$ into $f = f_1 f_2$ with $\operatorname{rank}(f_1) = k+1$ and $\operatorname{rank}(f_2) = n-k$.

$$
\begin{array}{l}
\\
k+1 \left\{ \right. \\
\\
\\
\\
\\
\left. \right\} n-k
\end{array}
\overbrace{\left(\begin{array}{ccc|c}
* & \dots & * & 0 \\
\vdots & & \vdots & \\
* & \dots & * & * \\
 & & \vdots & \vdots \\
 & * & * & * \\
\end{array}\right)}^{k+1}
\underbrace{\phantom{\begin{array}{ccc|c} * & \dots & * & 0 \end{array}}}_{n-k}
\tag{2.1.7}
$$

with $*$ any linear entry in $\mathbb{K}\langle X\rangle$, not all zero.

Remark 2.1.8. Notice that we consider the reducibility over the *algebraic closure* of the ground field, denoted by $\overline{\mathbb{K}}\langle X\rangle$.

A non-trivial ideal does not guarantee a factorization over $\mathbb{K}$ as seen in the following Example unless the field is algebraic closed.

Example 2.1.9. Let $f = x^2 - 2 \in \mathbb{K}\langle X\rangle$. Then the ideal $[b_{23} + a_{12}, a_{12}^2 - 2]$ is not trivial. For details see Example 2.3.2. If $\mathbb{K} = \mathbb{Q}$, then f is irreducible (i.e., if $f = f_1 f_2$, then either f_1 or f_2 is a unit), since $\sqrt{2} \notin \mathbb{Q}$. If $\mathbb{K} = \mathbb{R}$, then there is a pre-standard admissible transformation (P, Q) and therefore $f = \left(x - \sqrt{2}\right)\left(x + \sqrt{2}\right) = \left(x + \sqrt{2}\right)\left(x - \sqrt{2}\right)$ in $\mathbb{R}\langle X\rangle$.

Remark 2.1.10. In Theorem 3.1.1 the matrices P and Q are restricted to have determinant 1. For simplicity we treat the matrices P and Q as upper triangular matrices with ones on the diagonal in Proposition 2.1.6 (see (2.1.4)) to satisfy invertibility. Therefore the generation of the ideals is easier. As we can see in Section 2.4 we are able to check non-commutative reducible polynomials up to rank 17 and irreducible polynomials up to rank 12. In contrast, testing fullness of matrices of dimension 5 already can defeat the computer.

In Theorem 3.1.1 we use a trivial ideal for determining if a matrix is full, but it makes no difference whether the matrix is over $\mathbb{K}\langle X\rangle$ or over $\overline{\mathbb{K}}\langle X\rangle$. Searching for the factorization of a polynomial it is important, that the polynomial can factor over $\overline{\mathbb{K}}\langle X\rangle$, but not necessarily over $\mathbb{K}\langle X\rangle$.

Remark 2.1.11. For any polynomial $f \in \mathbb{K}\langle X\rangle$ with $\operatorname{rank}(f) = n \geq 2$ we can transform any minimal pre-standard admissible linear system $\pi = (1, A, \lambda)$ into an admissible linear system of the form $(1, A', 1)$ by dividing the last row by λ and multiplying the last column by λ.

Example 2.1.12. Let $f = x^3 - 10x^2 + 31x - 30 \in \mathbb{K}\langle X\rangle$. A minimal pre-standard admissible linear system of f is given by

$$\begin{pmatrix} 1 & -x & . & \frac{30}{31} - x \\ . & 1 & -x & \frac{10}{31}x \\ . & . & 1 & -\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 31 \end{pmatrix}, \quad s = \begin{pmatrix} f \\ x^2 - 10x \\ x \\ 31 \end{pmatrix}. \tag{2.1.13}$$

As mentioned in Remark 2.1.11 we can transform (2.1.13) into the following admissible linear system of the form $(1, A', 1)$:

$$\begin{pmatrix} 1 & -x & . & 30 - 31x \\ . & 1 & -x & 10x \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix} s = \begin{pmatrix} . \\ . \\ . \\ 1 \end{pmatrix}, \quad s = \begin{pmatrix} f \\ x^2 - 10x \\ x \\ 1 \end{pmatrix}. \tag{2.1.14}$$

For univariate polynomials, i.e., polynomials in one variable, the companion matrix provides a minimal linear representation. Further details about companion matrices and their usage for the factorization are given in [Sch17b, Section 3].

2.2 Implementation in FriCAS

The program code, which is based on Theorem 2.1.3 and Proposition 2.1.6, is written in FriCAS, a computer algebra system which is a descendant of Axiom. The program is based on several `.spad` files for the handling of non-commutative polynomials. Notice that these `.spad` files are not in the standard installation from FriCAS, but implemented by Konrad Schrempf who provided me with the files. The program codes are explained in detail in [Jan18, Section 2] and are used in the following examples of non-commutative polynomials.

Non-commutative polynomials, for example the polynomial $f = xy - 2yx + 3$, are defined in FriCAS with the package `ncpoly.spad` as follows:

```
f:NCP := x*y - 2*y*x + 3
```

Remark 2.2.1. In the package `ncpoly.spad` the macro `NCP` stands for `NonCommutativePolynomial(OrderedVariableList,Field)` with the ordered variable list `OrderedVariableList` (in our case X) and the field `Field` (in our case the

commutative field $\mathbb{K}$ as complex rational numbers). Notice that a non-commutative polynomial (defined as `NCP`) is represented by a minimal pre-standard admissible linear system.

Remark 2.2.2. In FriCAS we have to convert the entries of the system matrix A of the minimal linear representation $\pi = (u, A, v)$ to *non-commutative* polynomials with *commutative* polynomials as coefficients. The converted system matrix is denoted by L.

The function call is `Polyfact(f)` with f the non-commutative polynomial, represented by a minimal pre-standard admissible linear system, for testing. Depending on the rank of the polynomial different procedures are executed.

- *Polynomials of rank smaller than* 3 are irreducible (by definition) and print an error message ("Error: rank too small").
- An *irreducible polynomial of rank* 3 produces the output "trivial ideal- irreducible polynomial" and then the trivial Gröbner basis.
- For *reducible polynomials of rank* 3 the solutions for the transformation variables are printed and then the Gröbner basis is returned.
- *Reducible polynomials of rank* 4 have the right upper zero blocks first, then the solutions for the transformation variables and at the end there is the Gröbner basis. The output "no solution" means that the handled ideal is trivial.
- *Irreducible polynomials of rank greater than* 4 produce the output "no solution" several times and then the Gröbner basis is empty, since all Gröbner bases are trivial and there do not exist any suitable transformation matrices.
- *Irreducible polynomials of rank* 4 have the same output as irreducible polynomials of rank greater than 4, but additionally there is a solution for the transformation variables. These transformation variables can be chosen arbitrarily, since FriCAS uses place-holders. In FriCAS place-holders start with % and are automatically generated variables, for example $\%E$ is a place-holder.
- For *reducible polynomials of rank greater than* 4 the right upper zero blocks are printed first (maybe sometimes "no solution") and at the end the Gröbner basis is returned.

2.3 Examples

In this section we show some examples of non-commutative polynomial factorization and illustrate the FriCAS code from [Jan18, Section 2].

2.3.1 Non-commutative Polynomials of rank 3

The following non-commutative polynomials of rank 3 are discussed here.

```
f1:NCP := 1 - x*y
f2:NCP := y*y - 9
f21:NCP := x*x - 2
f22:NCP := 4*x*x - 9
```

Example 2.3.1. A minimal admissible linear system for the polynomial $f_2 := y^2 - 9$ is

$$\begin{bmatrix} 1 & -y & 9 \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad s = \begin{pmatrix} y^2 - 9 \\ y \\ 1 \end{pmatrix}.$$

The list of ideals is taken from the upper right corner of the matrix

$$P \cdot L(f_2) \cdot Q = \begin{pmatrix} 1 & a_{1,2} - y & a_{1,2}b_{2,3} + 9 + (-b_{2,3} - a_{1,2})\, y \\ 0 & 1 & b_{2,3} - y \\ 0 & 0 & 1 \end{pmatrix}$$

with P and Q of the form (2.1.4).

In this case we want to have a 1×1 upper right zero block, so the $(1,3)$-entry of the matrix $PL(f_2)Q$ should be zero. Therefore we get from the function `listIdealsf` the ideal generated by $[[a_{1,2}b_{2,3} + 9,\ -b_{2,3} - a_{1,2}]]$. The Gröbner basis of this ideal reads $\left[b_{2,3} + a_{1,2},\ a_{1,2}^2 - 9\right]$.

If we call the function `Polyfact(f2)` the output is

$$[[a_{1,2} = 3,\ b_{2,3} = -3],\ [a_{1,2} = -3,\ b_{2,3} = 3]]$$
$$\left[b_{2,3} + a_{1,2},\ a_{1,2}^2 - 9\right].$$

The first line gives us the two solutions for our transformation matrices, the second line shows us the Gröbner basis of this ideal. If we insert these two solutions into the matrix $PL(f_2)Q$, we receive the following two possible factorizations of the polynomial:

$$\begin{pmatrix} 1 & 3-y & 0 \\ . & 1 & -3-y \\ . & . & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & -3-y & 0 \\ . & 1 & 3-y \\ . & . & 1 \end{pmatrix},$$

i.e., $f_2 := y^2 - 9 = (y-3)(y+3) = (y+3)(y-3)$.

Example 2.3.2. Considering the polynomial $f_{21} := x^2 - 2$ we have the minimal admissible linear system

$$\begin{bmatrix} 1 & -x & 2 \\ 0 & 1 & -x \\ 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad s = \begin{pmatrix} x^2 - 2 \\ x \\ 1 \end{pmatrix}.$$

Calling the function `Polyfact(f21)` yields

$$\left[\left[a_{1,2} = -b_{2,3},\, b_{2,3}^2 - 2 = 0\right]\right]$$
$$\left[b_{2,3} + a_{1,2},\, a_{1,2}^2 - 2\right].$$

There is no solution over the field of (complex) rational numbers and this polynomial is irreducible over $\mathbb{Q}$, but over $\mathbb{R}$ we can factor this polynomial into $(x - \sqrt{2})(x+\sqrt{2}) = (x + \sqrt{2})(x - \sqrt{2})$.

Example 2.3.3. The polynomial $f_1 := 1 - xy$ with the minimal admissible linear system

$$\begin{bmatrix} 1 & -x & 1 \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad s = \begin{pmatrix} 1 - xy \\ -y \\ -1 \end{pmatrix}$$

delivers the output of the function `Polyfact(f1)`

"trivial ideal- irreducible polynomial"
$$[1],$$

which means that f_1 is an irreducible polynomial due to a trivial ideal. As the second line shows, the Gröbner basis is indeed trivial.

Example 2.3.4. The polynomial $f_{22} := 4x^2 - 9$ is represented by the minimal admissible linear system

$$\begin{bmatrix} 1 & -4x & 9 \\ 0 & 1 & -x \\ 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad s = \begin{pmatrix} 4x^2 - 9 \\ x \\ 1 \end{pmatrix}.$$

From the matrix

$$P \cdot L(f_{22}) \cdot Q = \begin{pmatrix} 1 & a_{1,2} - 4x & a_{1,2}b_{2,3} + 9 + (-4b_{2,3} - a_{1,2})\, x \\ 0 & 1 & b_{2,3} - x \\ 0 & 0 & 1 \end{pmatrix}$$

with the transformation matrices P and Q of the form (2.1.4) the function `Polyfact(f22)` produces the following output

$$\left[\left[a_{1,2} = 6,\, b_{2,3} = -\frac{3}{2}\right],\, \left[a_{1,2} = -6,\, b_{2,3} = \frac{3}{2}\right]\right]$$
$$\left[b_{2,3} + \frac{1}{4}a_{1,2},\, a_{1,2}^2 - 36\right].$$

Inserting these solutions in P and Q respectively we get the following two matrices

$$P_1 L Q_1 = \begin{pmatrix} 1 & 6 - 4x & 0 \\ . & 1 & -\frac{3}{2} - x \\ . & . & 1 \end{pmatrix} \quad \text{and} \quad P_2 L Q_2 = \begin{pmatrix} 1 & -6 - 4x & 0 \\ . & 1 & \frac{3}{2} - x \\ . & . & 1 \end{pmatrix},$$

which correspond to the factorizations $(4x - 6)\left(x + \frac{3}{2}\right)$ and $(4x + 6)\left(x - \frac{3}{2}\right)$.

2.3.2 Non-commutative Polynomials of rank 4

In this part we consider the following examples of non-commutative polynomials of rank 4 :

```
f31:NCP := x*y*z - 3*x*y + x*z + 2*y*z - 3*x - 6*y + 2*z - 6
f35:NCP := x*x*x - 10*x*x + 31*x - 30
f38:NCP := x - x*y*x
f4:NCP := x*y + y*x
```

The polynomial f_{35} is the same as in Example 2.1.12. Here we will investigate it on the basis of the FriCAS code in [Jan18, Section 2].

Remember that for non-commutative polynomials of rank 4 we first determine the minimal ranks of the factors and then we solve the system of equations of the ideal for the variables of the transformation matrices.

Example 2.3.5. The (non-commutative) polynomial $f_{35} := x^3 - 10x^2 + 31x - 30$ is represented by a minimal admissible linear system

$$\begin{bmatrix} 1 & -x & 0 & \frac{30}{31} - x \\ 0 & 1 & -x & \frac{10}{31}x \\ 0 & 0 & 1 & -\frac{1}{31}x \\ 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 31 \end{bmatrix}, \quad s = \begin{pmatrix} x^3 - 10x^2 + 31x - 30 \\ x^2 - 10x \\ x \\ 31 \end{pmatrix}.$$

To determine the minimal ranks of the factors we investigate the matrix

$$P \cdot L(f_{35}) \cdot Q = \begin{pmatrix} 1 & a_{1,2} - x & a_{1,2}b_{2,3} + a_{1,3} + (-b_{2,3} - a_{1,2})\,x & a_{1,3}b_{3,4} + a_{1,2}b_{2,4} + \frac{30}{31} + \left(-a_{1,2}b_{3,4} - b_{2,4} - \frac{1}{31}a_{1,3} + \frac{10}{31}a_{1,2} - 1\right)x \\ 0 & 1 & b_{2,3} + a_{2,3} - x & a_{2,3}b_{3,4} + b_{2,4} + \left(-b_{3,4} - \frac{1}{31}a_{2,3} + \frac{10}{31}\right)x \\ 0 & 0 & 1 & b_{3,4} - \frac{1}{31}x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with P and Q of the form (2.1.4).

First of all, we test whether there is a factor of rank 2, i.e., whether the Gröbner basis of the ideal generated by the coefficients of entries $(1,3)$ and $(1,4)$ is non-trivial (illustrated in (2.3.6)).

$$\left(\begin{array}{cc|cc} * & * & ? & ? \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{array}\right) \tag{2.3.6}$$

This Gröbner basis $\left[b_{3,4} + \left(\frac{1}{30}a_{1,2}^2 - \frac{1}{3}a_{1,2} + \frac{31}{30}\right) b_{2,4} + \frac{1}{30}a_{1,2}^2 - \frac{28}{93}a_{1,2} + \frac{661}{930},\ b_{2,3} + a_{1,2},\ a_{1,3} - a_{1,2}^2, a_{1,2}^3 - 10a_{1,2}^2 + 31a_{1,2} - 30\right]$ is indeed not trivial, therefore we have found an 1×2 right upper zero block and our first factor has rank 2. (Replace in (2.3.6) the question marks by zeros.)

Now we assume in the second row again a factor of rank 2 and test the Gröbner basis from before combined with the ideal generated by the entries $(1, 4)$ and $(2, 4)$ (illustrated in (2.3.7)).

$$\begin{pmatrix} * & * & \underline{\vert\, 0} & \underline{\vert\, ?} \\ * & * & * & \underline{\vert\, ?} \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \tag{2.3.7}$$

This Gröbner basis $\left[b_{3,4} + \frac{1}{31}a_{2,3} - \frac{10}{31},\ b_{2,4} - \frac{1}{31}a_{1,2}a_{2,3} + \frac{1}{31}a_{1,2}^2 + 1,\ b_{2,3} + a_{1,2},\ a_{2,3}^2 + \left(-a_{1,2} - 10\right) a_{2,3} + a_{1,2}^2 + 31,\ a_{1,3} - a_{1,2}^2,\ a_{1,2}^3 - 10a_{1,2}^2 + 31a_{1,2} - 30\right]$ is again non-trivial, so we have found the next factor of rank 2 and a further 2×1 right upper zero block. For this polynomial we are finished determining the minimal ranks, since we treated the last possible row. Therefore our factorized block form is

$$\begin{pmatrix} * & * & \underline{\vert\, 0} & 0 \\ * & * & * & \underline{\vert\, 0} \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$

With the Gröbner basis from these two right upper zero blocks we can solve for the variables of the two transformation matrices.

The function `Polyfact(f35)` yields the following output

$$1 \times 2$$
$$2 \times 1$$

$$\left[a_{1,2} = 2,\ a_{1,3} = 4,\ a_{2,3} = 5,\ b_{2,3} = -2,\ b_{2,4} = -\frac{25}{31},\ b_{3,4} = \frac{5}{31}\right],$$
$$\left[a_{1,2} = 3,\ a_{1,3} = 9,\ a_{2,3} = 5,\ b_{2,3} = -3,\ b_{2,4} = -\frac{25}{31},\ b_{3,4} = \frac{5}{31}\right],$$
$$\left[a_{1,2} = 2,\ a_{1,3} = 4,\ a_{2,3} = 7,\ b_{2,3} = -2,\ b_{2,4} = -\frac{21}{31},\ b_{3,4} = \frac{3}{31}\right],$$
$$\left[a_{1,2} = 5,\ a_{1,3} = 25,\ a_{2,3} = 7,\ b_{2,3} = -5,\ b_{2,4} = -\frac{21}{31},\ b_{3,4} = \frac{3}{31}\right],$$
$$\left[a_{1,2} = 3,\ a_{1,3} = 9,\ a_{2,3} = 8,\ b_{2,3} = -3,\ b_{2,4} = -\frac{16}{31},\ b_{3,4} = \frac{2}{31}\right],$$
$$\left[a_{1,2} = 5,\ a_{1,3} = 25,\ a_{2,3} = 8,\ b_{2,3} = -5,\ b_{2,4} = -\frac{16}{31},\ b_{3,4} = \frac{2}{31}\right]$$

$$\left[b_{3,4} + \frac{1}{31}a_{2,3} - \frac{10}{31},\ b_{2,4} - \frac{1}{31}a_{1,2}a_{2,3} + \frac{1}{31}a_{1,2}^2 + 1,\ b_{2,3} + a_{1,2},\right.$$
$$\left.a_{2,3}^2 + (-a_{1,2} - 10)\,a_{2,3} + a_{1,2}^2 + 31,\ a_{1,3} - a_{1,2}^2,\ a_{1,2}^3 - 10a_{1,2}^2 + 31a_{1,2} - 30\right].$$

The first two lines show us our right upper zero blocks, then the six possible solutions for the transformation matrices are printed and in the last line the Gröbner basis of the ideal of the zero blocks is returned.

These six solutions provide the following six factorizations:

$$\begin{pmatrix} 1 & 2-x & 0 & 0 \\ . & 1 & 3-x & 0 \\ . & . & 1 & \frac{5}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix}, \begin{pmatrix} 1 & 3-x & 0 & 0 \\ . & 1 & 2-x & 0 \\ . & . & 1 & \frac{5}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 2-x & 0 & 0 \\ . & 1 & 5-x & 0 \\ . & . & 1 & \frac{3}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix}, \begin{pmatrix} 1 & 5-x & 0 & 0 \\ . & 1 & 2-x & 0 \\ . & . & 1 & \frac{3}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 3-x & 0 & 0 \\ . & 1 & 5-x & 0 \\ . & . & 1 & \frac{2}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix}, \begin{pmatrix} 1 & 5-x & 0 & 0 \\ . & 1 & 3-x & 0 \\ . & . & 1 & \frac{2}{31}-\frac{1}{31}x \\ . & . & . & 1 \end{pmatrix}.$$

Since the admissible linear system of the polynomial $f_{35} := x^3 - 10x^2 + 31x - 30$ is pre-standard, we have to divide the last row by 31 and multiply the last column by 31 (see Remark 2.1.11) to receive the factorizations of the polynomial. Therefore we have the following six factorizations of the polynomial $f_{35} := x^3 - 10x^2 + 31x - 30$:

$$(x-2)(x-3)(x-5),\ (x-3)(x-2)(x-5),$$
$$(x-2)(x-5)(x-3),\ (x-5)(x-2)(x-3),$$
$$(x-3)(x-5)(x-2),\ (x-5)(x-3)(x-2).$$

Notice that we have in each factorization matrix an 1×2 and a 2×1 right upper zero block.

Example 2.3.8. A minimal admissible linear system for the polynomial $f_{38} := x - xyx$ is given by

$$\begin{bmatrix} 1 & -x & 0 & x \\ 0 & 1 & -y & 0 \\ 0 & 0 & 1 & -x \\ 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}, \quad s = \begin{pmatrix} x - xyx \\ -yx \\ -x \\ -1 \end{pmatrix}.$$

Due to the matrix

$$P \cdot L(f_{38}) \cdot Q =$$
$$\begin{pmatrix} 1 & a_{1,2} - x & a_{1,2}b_{2,3} + a_{1,3} - a_{1,2}y - b_{2,3}x & a_{1,3}b_{3,4} + a_{1,2}b_{2,4} - a_{1,2}b_{3,4}y + \\ & & & (-b_{2,4} - a_{1,3} + 1)\, x \\ 0 & 1 & b_{2,3} + a_{2,3} - y & a_{2,3}b_{3,4} + b_{2,4} - b_{3,4}y - a_{2,3}x \\ 0 & 0 & 1 & b_{3,4} - x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with the transformation matrices P and Q of the form (2.1.4) the output of the function `Polyfact(f38)` is

$$1 \times 2$$
$$\text{“no solution”}$$
$$[[a_{1,2} = 0,\ a_{1,3} = 0,\ a_{2,3} = \%E,\ b_{2,3} = 0,\ b_{2,4} = 1,\ b_{3,4} = \%F]]$$

$$[b_{2,4} - 1,\ b_{2,3},\ a_{1,3},\ a_{1,2}]$$

Therefore the decomposition of the polynomial $f_{38} := x - xyx$ has the following form

$$\left(\begin{array}{c|c|cc} * & * & 0 & 0 \\ \hline * & * & * & * \\ \hline * & * & * & * \\ * & * & * & * \end{array}\right). \tag{2.3.9}$$

In (2.3.9) we can easily see that the first factor of the polynomial has rank 2 and the second factor has rank 3.

The second line of the output ("no solution") refers to a trivial ideal of the PLQ-entries $(1,3),(1,4)$ and $(2,4)$. The solution for the transformation matrices contains for the variables $a_{2,3}$ and $b_{3,4}$ some place-holder (in FriCAS place-holders start with % and they are automatically generated variables, here $\%E$ and $\%F$). In the last line the Gröbner basis is returned.

With the solution for the transformation matrices we get the factorization

$$\begin{pmatrix} 1 & -x & 0 & 0 \\ . & 1 & a_{2,3}-y & a_{2,3}b_{3,4}+1-b_{3,4}y-a_{2,3}x \\ . & . & 1 & b_{3,4}-x \\ . & . & . & 1 \end{pmatrix}.$$

Let $a_{2,3}=b_{3,4}=0$. Then the matrix PLQ is

$$\begin{pmatrix} 1 & -x & 0 & 0 \\ . & 1 & -y & 1 \\ . & . & 1 & -x \\ . & . & . & 1 \end{pmatrix}$$

and we have the factorization $x(1-yx)$. If we insert any number for $a_{2,3}$ and $b_{3,4}$ we also get the second factor $(1-yx)$, since

$$a_{2,3}b_{3,4}+1-b_{3,4}y-a_{2,3}x-(a_{2,3}-y)(b_{3,4}-x)=1-yx.$$

Example 2.3.10. The polynomial $f_{31}:=xyz-3xy+xz+2yz-3x-6y+2z-6$ is represented by the minimal admissible linear system

$$\begin{bmatrix} 1 & -x & -2y & 3-z+3y+\frac{3}{2}x \\ 0 & 1 & -y & -\frac{1}{2}z+\frac{3}{2}y \\ 0 & 0 & 1 & -\frac{1}{2}z \\ 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

with the solution vector

$$s=\begin{pmatrix} xyz-3xy+xz+2yz-3x-6y+2z-6 \\ z-3y+yz \\ z \\ 2 \end{pmatrix}.$$

The function `Polyfact(f31)` returns

$$1 \times 2$$

$$2 \times 1$$

$$\left[\left[a_{1,2} = -2,\ a_{1,3} = 0,\ a_{2,3} = -1,\ b_{2,3} = 0,\ b_{2,4} = \frac{3}{2},\ b_{3,4} = \frac{3}{2}\right]\right]$$

$$\left[b_{3,4} - \frac{3}{2},\ b_{2,4} - \frac{3}{2},\ b_{2,3},\ a_{2,3} + 1,\ a_{1,3},\ a_{1,2} + 2\right],$$

so the factorized block form is

$$\left(\begin{array}{cc|cc} * & * & 0 & 0 \\ * & * & * & 0 \\ \hline * & * & * & * \\ * & * & * & * \end{array}\right),$$

i.e., three factors of rank 2 each.

If we now substitute this solution of the transformation matrices (of the form (2.1.4)) into the matrix

$P \cdot L(f_{31}) \cdot Q =$

$$\left(\begin{array}{cccc} 1 & a_{1,2} - x & \begin{array}{c} a_{1,2}b_{2,3} + a_{1,3} + \\ (-a_{1,2} - 2)\, y - b_{2,3}x \end{array} & \begin{array}{c} a_{1,3}b_{3,4} + a_{1,2}b_{2,4} + 3 + \\ \left(-\frac{1}{2}a_{1,3} - \frac{1}{2}a_{1,2} - 1\right) z + \\ \left((-a_{1,2} - 2)\, b_{3,4} + \frac{3}{2}a_{1,2} + 3\right) y + \left(-b_{2,4} + \frac{3}{2}\right) x \end{array} \\ 0 & 1 & b_{2,3} + a_{2,3} - y & a_{2,3}b_{3,4} + b_{2,4} + \left(-\frac{1}{2}a_{2,3} - \frac{1}{2}\right) z + \left(-b_{3,4} + \frac{3}{2}\right) y \\ 0 & 0 & 1 & b_{3,4} - \frac{1}{2}z \\ 0 & 0 & 0 & 1 \end{array}\right)$$

we get the system matrix

$$\left(\begin{array}{cccc} 1 & -2 - x & 0 & 0 \\ 0 & 1 & -1 - y & 0 \\ 0 & 0 & 1 & \frac{3}{2} - \frac{1}{2}z \\ 0 & 0 & 0 & 1 \end{array}\right).$$

Remember that this matrix corresponds to the factorization

$$(x - 2)(y - 1)(z - 3),$$

since we have to divide the last row by 2 and multiply the last column by 2 (see Remark 2.1.11).

Example 2.3.11. The polynomial $f_4 := xy + yx$ has rank 4 and a minimal admissible linear system is

$$\begin{bmatrix} 1 & -x & -y & 0 \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & -x \\ 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad s = \begin{pmatrix} xy + yx \\ y \\ x \\ 1 \end{pmatrix}.$$

The function `Polyfact(f4)` has the output

"no solution"
"no solution"
$[[a_{1,2} = \%L,\ a_{1,3} = \%K,\ a_{2,3} = \%J,\ b_{2,3} = \%I,\ b_{2,4} = \%H,\ b_{3,4} = \%G]]$

$[\,]$.

The first two lines correspond to non-trivial ideals, hence this non-commutative polynomial is irreducible.

Inserting the solution for the transformation matrices gives us the factorization matrix

$$PLQ = \begin{pmatrix} 1 & a_{1,2} - x & a_{1,2}b_{2,3} + a_{1,3} - y - b_{2,3}x & a_{1,3}b_{3,4} - a_{1,2}b_{2,4} + \\ & & & (-b_{3,4} - a_{1,2})\, y + (b_{2,4} - a_{1,3})\, x \\ 0 & 1 & b_{2,3} + a_{2,3} & a_{2,3}b_{3,4} + b_{2,4} - y - a_{2,3}x \\ 0 & 0 & 1 & b_{3,4} - x \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Similarly as in Section 2.3.2 in the example polynomial f_{38} (see Example 2.3.8) we can show that these operations do not change irreducibility.

2.3.3 Non-commutative Polynomials of rank greater than 4

For polynomials of rank greater than 4 we are only able to determine the minimal ranks of the factors, since the solution variety has positive dimension. In this section we investigate the following examples of non-commutative polynomials of rank greater than 4 :

```
f5:NCP := 1 - 1*y*x - x*y + x*y^2*x
p1:NCP := 3*x - 2*x*y*x - x*y*x*y*x
q1:NCP := 9 - 9*x*y - x*y*x*y + x*y*x*y*x*y
g:NCP := 1 - x*y*z*y*x*z
```

Determining the minimal ranks of the factors works the same way as for polynomials of rank 4 (see Section 2.3.2). Starting with a factor of rank 2, we check the ideal of the coefficients of the PLQ-entries $(1,3)$, $(1,4)$, $\ldots$, $(1,n)$ with n the order of the matrix.

- If it is non-trivial, we found the first factor of minimal rank 2 and start in the second row again with a factor of rank 2, checking the ideal of entries $(2,4)$, $\ldots$, $(2,n)$ combined with the previous ideal.

- If the ideal is trivial, we assume a factor of rank 3 and check the ideal generated by the coefficients of the PLQ-entries $(1,4)$, $\ldots$, $(1,n)$, $(2,4)$, $\ldots$, $(2,n)$.

This will be repeated until we handled the $(n-2)$-row.

Example 2.3.12. The polynomial $f_5 := 1 - yx - xy + xy^2x$ of rank 5 is represented by the following minimal admissible linear system

$$\begin{bmatrix} 1 & -x & 0 & y & -1 \\ 0 & 1 & -y & 0 & y \\ 0 & 0 & 1 & -y & 0 \\ 0 & 0 & 0 & 1 & -x \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad s = \begin{pmatrix} 1 - yx - xy + xy^2x \\ -y + y^2x \\ yx \\ x \\ 1 \end{pmatrix}.$$

In the function `Polyfact(f5)` we use the matrix $PL(f_5)Q =$

$$\begin{pmatrix} 1 & a_{1,2} - x & a_{1,2}b_{2,3} + a_{1,3} - a_{1,2}y - b_{2,3}x & a_{1,3}b_{3,4} + a_{1,2}b_{2,4} + a_{1,4} + (-a_{1,2}b_{3,4} - a_{1,3} + 1)\,y - b_{2,4}x & a_{1,4}b_{4,5} + a_{1,3}b_{3,5} + a_{1,2}b_{2,5} - 1 + (-a_{1,3} + 1)\,b_{4,5}y + (-a_{1,2}b_{3,5} + a_{1,2})\,y + (-b_{2,5} - a_{1,4})\,x \\ 0 & 1 & b_{2,3} + a_{2,3} - y & a_{2,3}b_{3,4} + b_{2,4} + a_{2,4} + (-b_{3,4} - a_{2,3})\,y & a_{2,4}b_{4,5} + a_{2,3}b_{3,5} + b_{2,5} + (-a_{2,3}b_{4,5} - b_{3,5} + 1)\,y - a_{2,4}x \\ 0 & 0 & 1 & b_{3,4} + a_{3,4} - y & a_{3,4}b_{4,5} + b_{3,5} - b_{4,5}y - a_{3,4}x \\ 0 & 0 & 0 & 1 & b_{4,5} - x \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

with the transformation matrices P and Q of the form (2.1.4) to determine the ideals and therefore finding the minimal ranks of the factors. With `Polyfact(f5)` we get

the output

"no solution"
2×2
"no solution"

$$[a_{1,4}b_{4,5} + b_{3,5} - 1,\ (a_{1,3} - 1)\, b_{4,5} + a_{12}b_{3,5} - a_{1,2},\ b_{3,5}^2 - 2b_{3,5} + 1,\ a_{1,4}b_{3,5} - a_{1,4},$$
$$(a_{1,3} - 1)\, b_{3,5} - a_{1,3} + 1,\ b_{3,4} + a_{1,4},\ b_{2,5} + a_{1,4},\ b_{2,4},\ a_{2,4},\ a_{2,3} - a_{1,4},\ a_{1,4}^2,\ (a_{1,3} - 1)\, a_{1,4},$$
$$a_{1,2}a_{1,4} - a_{1,3} + 1,\ a_{1,3}^2 - 2a_{1,3} + 1].$$

Therefore we see that the first and third treated ideal, which concern the entries $(1,3)$, $(1,4)$, $(1,5)$ respectively $(1,4)$, $(1,5)$, $(2,4)$, $(2,5)$, $(3,5)$, are trivial ("no solution").

The decomposition of the non-commutative polynomial $f_5 := 1 - yx - xy + xy^2x$ into two factors of rank 3 looks like

$$\left(\begin{array}{ccc|cc} * & * & * & 0 & 0 \\ * & * & * & 0 & 0 \\ \hline * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array}\right).$$

Indeed $1 - yx - xy + xy^2x = (1 - xy)(1 - yx)$ has two irreducible factors of rank 3.

Example 2.3.13. The following minimal admissible linear system represents the non-commutative polynomial $p_1 := 3x - 2xyx - xyxyx$ of rank 6

$$\begin{bmatrix} 1 & -2x & 0 & 0 & 0 & 3x \\ 0 & 1 & -y & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{2}x & 0 & -x \\ 0 & 0 & 0 & 1 & -y & 0 \\ 0 & 0 & 0 & 0 & 1 & -x \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

with solution vector

$$s = \begin{pmatrix} 3x - 2xyx - xyxyx \\ -yx - \frac{1}{2}yxyx \\ -x - \frac{1}{2}xyx \\ -yx \\ -x \\ -1 \end{pmatrix}.$$

For this polynomial the function `Polyfact(p1)` produces the following output

$$1 \times 4$$
"no solution"
$$3 \times 2$$
"no solution"
$$\left[b_{5,6} + \left(-\frac{4}{3} a_{2,4} + \frac{4}{3} \right) b_{3,6},\ b_{4,6} + 2a_{2,4} + 2,\ b_{4,5},\ b_{3,5} + a_{2,4},\ b_{2,6} - \frac{3}{2},\right.$$
$$\left. b_{2,5},\ b_{2,4},\ b_{2,3},\ a_{3,5} - a_{2,4},\ a_{3,4},\ a_{2,5} - a_{2,3}a_{2,4},\ a_{2,4}^2 + a_{2,4} - \frac{3}{4},\ a_{1,5},\ a_{1,4},\ a_{1,3},\ a_{1,2} \right].$$

The ideals corresponding to the first two factors to be of rank 2 and the ideal with the last factor rank 2 are trivial. The Gröbner basis concerning the zero blocks is non-trivial, so the non-commutative polynomial is reducible. The "form" of the system matrix which shows the decomposition of the polynomial is

$$\left(\begin{array}{c|c|c|c|c|c} * & * & 0 & 0 & 0 & 0 \\ \hline * & * & * & * & 0 & 0 \\ \hline * & * & * & * & 0 & 0 \\ \hline * & * & * & * & * & * \\ \hline * & * & * & * & * & * \\ * & * & * & * & * & * \end{array} \right),$$

in which we can see a factor of rank 2 and two factors of rank 3.

Indeed the non-commutative polynomial $p_1 := 3x - 2xyx - xyxyx$ of rank 6 factors into $x(1 - yx)(3 + yx)$. Therefore the first factor has rank 2, the second and the third have rank 3.

Example 2.3.14. A minimal admissible linear system for the non-commutative polynomial $q_1 := 9 - 9xy - xyxy + xyxyxy$ of rank 7 is

$$\begin{bmatrix} 1 & -9x & 0 & 0 & 0 & 0 & -9 \\ 0 & 1 & -\frac{1}{9}y & 0 & 0 & 0 & y \\ 0 & 0 & 1 & -x & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -y & 0 & y \\ 0 & 0 & 0 & 0 & 1 & -x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -y \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

with

$$s = \begin{pmatrix} 9 - 9xy - xyxy + xyxyxy \\ -y - \frac{1}{9}yxy + \frac{1}{9}yxyxy \\ -xy + xyxy \\ -y + yxy \\ xy \\ y \\ 1 \end{pmatrix}.$$

The output of the function `Polyfact(q1)` is

$$\begin{aligned} &\text{“no solution”} \\ &2 \times 4 \\ &\text{“no solution”} \\ &4 \times 2 \\ &\text{“no solution”} \\ &\big[b_{6,7} + (-a_{3,5} + 1)\, b_{2,7},\ b_{5,7} + a_{3,5} - 1,\ b_{5,6},\ b_{4,7} + \big(a_{1,3}a_{3,5} - a_{1,3}^2 + 9\big)\, b_{2,7}, \\ &b_{4,6} + a_{3,5},\ b_{4,5} + \big(-a_{1,3}^2 + a_{1,3} + 9\big)\, b_{2,5},\ b_{3,7} - a_{1,3}a_{3,5} + a_{1,3}^2 - 9,\ b_{3,6},\ b_{3,5} + a_{1,3}, \\ &b_{3,4},\ b_{2,6} - \frac{1}{9}a_{1,3}a_{3,5} + \frac{1}{9}a_{1,3}^2,\ b_{2,4} + \frac{1}{9}a_{1,3},\ a_{4,6} - a_{3,5},\ a_{4,5},\ a_{3,6} - a_{3,4}a_{3,5}, \\ &a_{3,5}^2 + (-a_{1,3} - 1)\, a_{3,5} + a_{1,3}^2 - 9,\ a_{2,6} - \frac{1}{9}a_{1,3}^2,\ a_{2,5},\ a_{2,4} - \frac{1}{9}a_{1,3},\ a_{2,3}, \\ &a_{1,6} - \frac{1}{9}a_{1,2}a_{1,3}^2,\ a_{1,5} - a_{1,3}^2, a_{1,4} - \frac{1}{9}a_{1,2}a_{1,3},\ a_{1,3}^3 - a_{1,3}^2 - 9a_{1,3} + 9\big]. \end{aligned}$$

The system matrix of this non-commutative polynomial has the form

$$\left(\begin{array}{cc|c|cc|cc} * & * & * & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 \\ \hline * & * & * & * & * & 0 & 0 \\ * & * & * & * & * & 0 & 0 \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \end{array}\right),$$

so there are three factors of rank 3.

Indeed the non-commutative polynomial $q_1 := 9 - 9xy - xyxy + xyxyxy$ factors into $(1 - xy)(3 - xy)(3 + xy)$. So we have three factors of rank 3, which we can see in the decomposition.

Example 2.3.15. For the irreducible non-commutative polynomial $g := 1 - xyzyxz$ the function `Polyfact(g)` yields

"no solution"
"no solution"
"no solution"
"no solution"
"no solution"
[].

This output shows us that all ideals are trivial and therefore this polynomial is irreducible.

2.4 Observations

With the program code in FriCAS, which is explained in detail in [Jan18, Section 2], we are able to handle *reducible* non-commutative polynomials up to rank 17 to determine the minimal ranks of their factors. For non-commutative polynomials of rank 2 and 3 we are able to give explicit solution(s) of the factorization of the polynomials, for non-commutative polynomials of rank greater than 3 we determine the minimal ranks of the factors. Giving explicit solutions of the factorization is out of scope.

A polynomial of rank 18 leads to an error "Control stack exhausted", i.e., there is no more place for function calls (see Section 1.4). For *irreducible* non-commutative polynomials of rank 12 we get a solution of irreducibility, whereas an irreducible non-commutative polynomial of rank 13 produces an error "Control stack exhausted".

Remark 2.4.1. Our treated non-commutative polynomials consists of factors of rank at most 3, i.e., the factor has the form $(a \pm bc)$ with $a \in \mathbb{Z}$ and $b, c \in X$.

Remark 2.4.2. In the program code for determining the minimal ranks of the factors in [Jan18, Section 2] we add in each step the previous Gröbner basis of the coefficients to the current Gröbner basis of the coefficients. If we add the original generating set of the previous *ideal* instead of the previous *Gröbner basis* to the current Gröbner basis, we are only able to handle *reducible* non-commutative polynomials up to rank 14 and *irreducible* non-commutative polynomials up to rank 11 without any error.

The run-time adding the previous ideal does not differ from the run-time adding the previous Gröbner basis for *reducible* non-commutative polynomials until rank 10 and for *irreducible* non-commutative polynomials until rank 11. Afterwards the run-time adding the previous ideal is increasing rapidly.

Example 2.4.3. The *reducible* non-commutative polynomial $f_{10} = (1-xy)(2+yx)(3-yz)(2-zy)(1-xz)$ of rank 11 needs approximately one second as total run-time (adding

the previous Gröbner basis), whereas the total run-time adding the previous ideal is approximately 3 seconds.

Example 2.4.4. The run-time (adding the previous ideal) of the *reducible* non-commutative polynomial $f_{13} = (1 - xy)(2 + yx)(3 - yz)(2 - zy)(1 - xz)(3 + zx)x$ of rank 14 is already about 6 minutes, whereas sensing the previous Gröbner basis reduces the run-time to approximately 10 seconds.

The run-time of *irreducible* non-commutative polynomials is not sensitive on adding the previous ideal or the previous Gröbner basis.

Non-commutative polynomials of rank less than or equal to 10 have (for calculating the factorization respectively determining the minimal ranks of the factors) a total runtime of less than one second.

Example 2.4.5. The *reducible* non-commutative polynomial $f_{14} = (1-xy)(2+yx)(3-yz)(2 - zy)(1 - xz)(3 + zx)xz$ of rank 15 has total run-time of about 22 seconds, whereas the polynomial $f_{15} = f_{14} * y$ of rank 16 needs already about 4 minutes as total run-time. The total run-time of the polynomial $f_{16} = f_{15} * (1 - x)$ of rank 17, our current rank limit for determining the minimal ranks of the factors without an error, is approximately 37 minutes.

If we increase the rank of the factors, we observe different run-times.

Example 2.4.6. The non-commutative polynomial $h_2 = (1-xyz)(1-zyx)(3+yxz)y$ of rank 11 has total run-time of about 12 seconds, whereas the polynomial f_{10} from Example 2.4.3 (also rank 11) needs approximately one second.

But there are even differences in the total run-time of one polynomial depending on the construction of the minimal admissible system of the polynomial.

Example 2.4.7. The polynomial $q_1 := 9-9xy-xyxy+xyxyxy$ of rank 7, constructed with the sum rule (1.2.19), has total run-time 0.22 seconds. The same polynomial, according to the product rule (1.2.20), $q := (1-xy)(3-xy)(3+xy)$ needs 0.06 seconds.

Remark 2.4.8. If we consider in each step the time which is needed only for the calculation of the Gröbner basis of an ideal, we discover that trivial Gröbner bases are calculated very quickly. As soon as we find the first factor of the polynomial the calculation time of the Gröbner basis is increasing. The calculation time of the Gröbner basis is always increasing, when finding a factor of the polynomial, but the first factor requires the major calculation time. This observation occurs for polynomials of rank greater than or equal to 7.

Example 2.4.9. For the *reducible* non-commutative polynomial $h_2 = (1 - xyz)(1 - zyx)(3+yxz)y$ we see in Table 2.1 the times which are needed to calculate the Gröbner bases of the corresponding zero block. If the Gröbner basis is non-trivial, we see the size of the zero block. Therefore we observe Remark 2.4.8.

calculation time of Gröbner basis	row × column of the zero block
0.009	0×0
0.028	0×0
12.005	$\mathbf{3 \times 7}$
0.058	0×0
0.066	0×0
0.222	$\mathbf{6 \times 4}$
0.076	$\mathbf{7 \times 3}$
0.037	0×0
0.018	0×0

Table 2.1: Calculation times of the Gröbner bases for the polynomial h_2 in seconds

3 Testing Fullness of matrices

Recall that a full matrix has no factorization into rectangular matrices of smaller dimension and is invertible over the free field. Full matrices are for example used to solve the word problem in the free field (see [Sch17a]) or are part of a linear representation of an element of the free field.

The main theorem in this chapter is Theorem 3.1.1, which characterizes full matrices and indicates an algorithm for testing. An analogue of this theorem is used for the factorization of non-commutative polynomials in Chapter 2 (see Theorem 2.1.3 and Proposition 2.1.6).

In Section 3.2 we explain some examples of full and non-full matrices and in Section 3.3 we describe the implementation in FriCAS. In Section 3.4 we collect some observations concerning the implementation in FriCAS and the FriCAS code, which is explained in detail in [Jan18, Section 3].

3.1 The main theorem

Deciding whether a linear square matrix over $\mathbb{K}\langle X\rangle$ is full or not is possible with the following theorem. Let L be such a matrix, i.e. $L = L_1 + \sum_{x\in X} L_x \otimes x$ with L_1 and L_x matrices over $\mathbb{K}$. Furthermore let P and Q be matrices of order n whose entries are commutative variables a_{ij} respectively b_{ij}. Denote $\mathbb{K}[a, b]$ as the corresponding $\mathbb{K}$-algebra in the variables a_{ij}, b_{ij}.

Theorem 3.1.1 ([CR99], Theorem 4.1)**.** *For each $r \in \{1, \ldots, n\}$, denote by I_r the ideal of $\mathbb{K}[a, b]$ generated by the polynomials $\det(P)-1$, $\det(Q)-1$ and the coefficients of each $x \in X \cup \{1\}$ in the (i, j) entries of the matrix PLQ for $1 \le i \le r$, $r \le j \le n$. Then the linear matrix L is full if and only if for all $r \in \{1, \ldots, n\}$ the ideal I_r is trivial.*

Remark 3.1.2. Due to a misprint, in [CR99] the coefficients of L_1 are omitted!

Proof. Let $\overline{\mathbb{K}}$ be the algebraic closure of $\mathbb{K}$. Consider the embedding $\mathbb{K}\langle X\rangle \to \overline{\mathbb{K}}\langle X\rangle$. From [CR99, Theorem 6.4.6] it follows that L is full if and only if L (considered as an element of $\overline{\mathbb{K}}\langle X\rangle^{n\times n}$) is full. By Lemma 1.1.11, L is not full, if and only if it is associated over $\mathbb{K}$ to a hollow matrix, i.e., if for some $r \in \{1, \ldots, n\}$, there exist invertible matrices P and Q (with determinant 1) over $\overline{\mathbb{K}}$ such that PLQ has a $r \times (n + 1 - r)$ zero submatrix in the upper right corner. This is equivalent to the

ideal I_r being non-trivial over $\overline{\mathbb{K}}$. So L is full if and only if for each r, the ideal I_r (in $\overline{\mathbb{K}}[a, b]$) is trivial, i.e., hence I_r contains 1. But therefore L is full if and only if the ideal I_r in $\mathbb{K}[a, b]$ is trivial for all r. □

Remark 3.1.3. Notice that in our case the entries of the matrix L are *non-commutative* polynomials. In contrast, the ideal I_r consists of polynomials in *commuting* variables a_{ij} and b_{ij}.

Therefore we can test if 1 belongs to the ideal of (commutative) polynomials given by a finite number of generators to decide whether the matrix is full or not. One possibility is the use of Gröbner bases.

3.2 Examples

In this section we comment on some examples of full and non-full matrices.

Example 3.2.1. Considering the matrix

$$R_1 := \begin{pmatrix} 1 & 0 & 0 \\ -x & 0 & 0 \\ y & x & z \end{pmatrix}$$

we indeed get the following result for testing the ideals in Theorem 3.1.1

$$[[1],\, [a_{1,1},\, (a_{1,2}a_{2,3} - a_{1,3}a_{2,2})\, a_{3,1} - 1,\, a_{2,1},\, (b_{1,2}b_{2,3} - b_{1,3}b_{2,2})\, b_{3,1} - 1,\, b_{3,2},\, b_{3,3}],\, [1]]$$

and the second ideal is not trivial. Additionally, the matrix is hollow and by Proposition 1.1.10 a 3×3 matrix cannot be full, if it contains a 2×2 zero submatrix, since $2 + 2 > 3$.

A non-full matrix factors into two matrices $\tilde{P} \in \mathbb{K}\langle X\rangle^{n\times p}$ and $\tilde{Q} \in \mathbb{K}\langle X\rangle^{p\times n}$ with $p < n$ such that $M = \tilde{P}\tilde{Q}$. In the present example the matrices

$$\tilde{P} = \begin{pmatrix} 1 & 0 \\ -x & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \tilde{Q} = \begin{pmatrix} 1 & 0 & 0 \\ y & x & z \end{pmatrix}$$

do the trick.

Example 3.2.2. The matrix

$$N_1 := \begin{pmatrix} 1 & x & 0 & 0 \\ 0 & 1 & x & -y \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

is full, since the ideals from Theorem 3.1.1 are trivial (Output: [[1], [1], [1], [1]]).

Example 3.2.3. The matrix

$$N_2 := \begin{pmatrix} 1 & -x & -y & 0 \\ 0 & 1 & 0 & -x \\ -z & 0 & -x & 0 \\ 0 & -z & 0 & 0 \end{pmatrix}$$

results in the four Gröbner bases

$$[[1], [1], [1], [1]].$$

Therefore all relevant ideals are trivial and the matrix is full.

3.3 Implementation in FriCAS

In the implementation in FriCAS of the fullness test, which is based on Theorem 3.1.1, the matrices P and Q have the following form

$$P = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}, \quad Q = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & & \ddots & \vdots \\ b_{n-1,1} & b_{n-1,2} & \dots & b_{n-1,n} \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix}. \tag{3.3.1}$$

We compute Gröbner bases of the ideals of the *commutative* polynomials of the coefficients in the matrix PLQ in Theorem 3.1.1. The complexity depends on the term order used.

We use the functions `full?, fullHDMP?, fullQP?` and `fullQPHDMP?`.

Remark. The function call is `full?(L)` with L the matrix for testing. In FriCAS for functions with one argument sometimes parentheses may be omitted. In this case we also can call the function with `full? L`. Analogously for the other functions. Notice that the function name includes the "?" and "?" is a legal letter for function name.

The functions `full?` and `fullQP?` use the lexicographical term order, the functions `fullHDMP?` and `fullQPHDMP?` use the reverse lexicographical term order. The functions `full?` and `fullHDMP?` for testing the fullness of a matrix are explained in detail in [Jan18, Section 3]. The function `fullQP?` (respectively `fullQPHDMP?`) is similar to the function `full?` (respectively `fullHDMP?`), it uses only a different term order. Therefore we only have to exchange P and Q in line 87 (respectively line 114) in [Jan18, Section 3].

Testing matrices for fullness depends, among other things, on the choice of the term order and on the order of the variables. We are not even able to guarantee a result with a certain term order up to a special dimension for matrices testing the fullness of the matrix. First of all, successful termination of the algorithm (instead of errors, see Section 1.4) depends on the values of the matrix entries and on the number of zero entries. Secondly, the size of intermediate results in the Gröbner bases computation is highly sensitive on the chosen term order. Another reason is the used order of the variables, which influences the output of the function and the run-time. The functions `full?` and `fullHDMP?` uses the order of the variables $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$, whereas the functions `fullQP?` and `fullQPHDMP?` use the order of the variables $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.

3.4 Observations

There are full matrices of rank 5 which result in a "Control stack exhausted" error using the (reverse) lexicographical term order, but there are also successful terminations of full matrices of rank 5 with the (reverse) lexicographical term order. Non-full matrices of dimension 5 may also result in a "Control stack exhausted" error. On the other hand there are non-full matrices of dimension 7 which terminate with the lexicographical term order in approximately 30 seconds. Testing a matrix of dimension higher or equal 5 will result in a "Control stack exhausted" error or will take sometimes much time. We conjecture that full matrices of rank 6 never terminate successfully in the current computing environment.

Example 3.4.1. Testing the matrix

$$M_1 := \begin{pmatrix} 1 & x & 0 & 0 & 0 & 0 \\ 0 & 1 & x & 0 & 0 & 0 \\ 0 & 0 & 1 & x & 0 & 0 \\ 0 & 0 & 0 & 1 & x & 0 \\ 0 & 0 & 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

returns for the lexicographical term order (with function `full?`) as well as for the reverse lexicographical term order (with function `fullHDMP?`) a "Control stack exhausted" error. The matrix

$$M_2 := \begin{pmatrix} 1 & x & 0 & 0 & 0 \\ 0 & 1 & x & 0 & 0 \\ 0 & 0 & 1 & x & 0 \\ 0 & 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

of rank 5 returns the trivial Gröbner bases with the lexicographical term order (with function `full?`) in approximately 18 minutes and with the reverse lexicographical term order (with function `fullHDMP?`) also a "Control stack exhausted" error.

Remark 3.4.2. In this Section we are talking about the dimension of a matrix, which may differ from the rank of the matrix. Especially for non-full matrices the dimension is greater or equal than the rank of the matrix. (For more information about the rank of a matrix see [FR04].)

In the following we experiment with different types of matrices and compare between different term and variable orders respectively to determine up to which dimension of the matrix calculations are feasible. We define $*$ as non-trivial matrix entries, i.e., the matrix entries are not all scalar and not all zero entries of the alphabet.

Example 3.4.3. A matrix of the form (3.4.4) with non-trivial first row and the other rows consisting of zeros is non-full. This matrix has rank 1 and the dimension is n.

$$\left.\begin{pmatrix} \overbrace{* & * & \dots & *}^{n} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}\right\} n \tag{3.4.4}$$

In Table 3.1 the runtime for matrices of dimension 5, 6 and 7 with the lexicographical and the reverse lexicographical term order are shown. Remember that the functions `full?` and `fullHDMP?` use the order of the variables $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$, whereas the functions `fullQP?` and `fullQPHDMP?` use the order of the variables $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$.

Observations (in Example 3.4.3 in dimension 7)**.** Testing such a matrix of dimension 7

- with the *lexicographical term order* and variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ results after approximately half a minute in non-trivial ideals, whereas

- with the *reverse lexicographical term order* the runtime is greater than 40 minutes (after 40 minutes it was cancelled by hand).

- The reverse lexicographical term order with variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ has total run-time of about 11 minutes, whereas

- for the same variable order but the lexicographical term order FriCAS turns into LDB.

Observations (in Example 3.4.3 in dimension 6)**.** For such a matrix of dimension 6 the order of the variables is very sensitive concerning the run-time. Using the order of the variables $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ we get a total run-time below one second, whereas only changing the order of variables to $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ increases the run-time to more than 1.5 minutes.

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 5$	0.12	0.77	0.63	0.21
$n = 6$	0.87	299.12	101.62	7.11
$n = 7$	27.2	$>$ 40 min	$\sim$ 20 min LDB	650.46

Table 3.1: Runtime for the matrix of the form (3.4.4) in seconds

Example 3.4.5. A matrix of the form

$$M = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ 2a_{1,1} & 2a_{1,2} & \ldots & 2a_{1,n} \\ \vdots & \vdots & \vdots & \vdots \\ na_{1,1} & na_{1,2} & \ldots & na_{1,n} \end{pmatrix} \tag{3.4.6}$$

with $a_{1,i}$, $i = 1, \ldots, n$ and $n = \dim M$, non-trivial entries of the alphabet is non-full. The rank of the matrix M is 1, the dimension is n.

Observation (in Example 3.4.5 in dimension 7)**.** Testing the fullness of the matrix M of dimension 7 with the *lexicographical term order* and variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ gives in approximately half an hour a result, whereas the others are cancelled by hand after several minutes.

Observations (in Example 3.4.5)**.** In Table 3.2 we see that for this matrix form independent from the dimension the lexicographical term order (function `full?`) with variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ is the best choice for a low total run-time. The function `fullQP?` with lexicographical term order and variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ has the major total run-time and the two functions with the reverse lexicographical term order are in between.

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 5$	0.29	1.74	5.18	0.82
$n = 6$	9.19	526.94	948.78	76.92
$n = 7$	1450.78	$>$ 45 min	$>$ 40 min	$>$ 30 min

Table 3.2: Runtime for matrices of the form (3.4.6) in seconds

Example 3.4.7. A matrix of the form

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,5} & 0 & \dots & 0 \\ 2a_{1,1} & \dots & 2a_{1,5} & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 5a_{1,1} & \dots & 5a_{1,5} & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \vdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix} \tag{3.4.8}$$

with $a_{1,i}$, $i = 1, \dots, 5$ non-trivial entries of the alphabet has dimension n, but its rank is 1. Matrices of this form of dimension 7 do not guarantee a result as seen in Table 3.3.

Observations (in Example 3.4.7 in dimension 5 and 6)**.** For matrices of dimension 5 and 6 we observe that the lexicographical term order with variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ has the lowest run-time. Switching the variable order to $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ results in this case in the highest run-time. The reverse lexicographical term orders lie in between, but the variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ is "faster" than the variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$.

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 5$	0.29	1.74	5.18	0.82
$n = 6$	5.88	324.64	651.35	41.95
$n = 7$	479.84	CSE	$\sim$ 48 min LDB	$>$ 40 min

Table 3.3: Runtime for matrices of the form (3.4.8) in seconds

Example 3.4.9. A matrix of the form (3.4.10) with the last row a zero row and the second to the $(n-1)$-th row shifting the previous row of one place to the left (σ_i for

$i = 1, \ldots, n-2$ is a cyclic permutation of one place to the left)

$$\begin{pmatrix} a_{1,1} & \ldots & a_{1,n} \\ \sigma_1(a_{1,1}) & \ldots & \sigma_1(a_{1,n}) \\ \vdots & & \vdots \\ \sigma_{n-2}(a_{1,1}) & \ldots & \sigma_{n-2}(a_{1,n}) \\ 0 & \ldots & 0 \end{pmatrix} \tag{3.4.10}$$

has dimension n and rank 1. This matrix with $a_{i,j}$ non-zero entries of the alphabet only has zeros in the last row. Therefore it raises an error of type CSE for dimension 5.

Observations (in Example 3.4.9)**.** The lexicographical term order with different variable orders and the reverse lexicographical term order with variable term order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ have almost the same total run-time (see Table 3.4). The minimal total run-time of approximately one second corresponds to the reverse lexicographical term order with variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 4$	4.71	4.06	5.64	1.24
$n = 5$	CSE	CSE	CSE	CSE
$n = 6$	CSE	CSE	CSE	CSE

Table 3.4: Runtime for matrices of the form (3.4.10) in seconds

Example 3.4.11. Testing the (non-full) matrix of the form (3.4.12) with non-trivial first row and non-trivial last column gives a result for matrix dimension 5.

$$\left.\begin{pmatrix} \overbrace{* & * & \ldots & *}^{n} \\ 0 & \ldots & 0 & * \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & * \end{pmatrix}\right\} n \tag{3.4.12}$$

Observations (in Example 3.4.11)**.** Testing a matrix of this form of dimension 6 or dimension 7 leads to an error CSE. For a matrix of dimension 5 the *lexicographical term order* (independent from the variable order) is "faster" than the *reverse lexicographical term order*. Table 3.5 shows the total run-time of the FriCAS codes from [Jan18, Section 3].

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 5$	142	198.39	137.47	158.34
$n = 6$	CSE	CSE	CSE	CSE
$n = 7$	CSE	CSE	CSE	CSE

Table 3.5: Runtime for the matrix of the form (3.4.12) in seconds

Example 3.4.13. Testing a matrix of the form (3.4.14)

$$\left.\begin{pmatrix} \overbrace{\begin{matrix} * & 0 & \dots & 0 \end{matrix}}^{n} \\ \begin{matrix} \vdots & \vdots & \ddots & \vdots \end{matrix} \\ \begin{matrix} * & 0 & \dots & 0 \end{matrix} \\ \begin{matrix} * & * & \dots & * \end{matrix} \end{pmatrix}\right\} n \tag{3.4.14}$$

with non-trivial first column and last row has for dimension 5 a total run-time of at most one minute. The *lexicographical term order* (independent of the variable order) and the *reverse lexicographical term order* with variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \dots, n$ and $j = 1, \dots, n$ are even under 30 seconds of total run-time. Matrices of this form of dimension 6 already lead to runtime exceeding one hour. (See Table 3.6.)

dimension	`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
$n = 5$	24.69	26.78	22.34	66.47
$n = 6$	> 60 min	> 30 min	> 90 min	> 30 min
$n = 7$	> 40 min	> 30 min	~ 28 min LDB	> 130 min

Table 3.6: Runtime for the matrix of the form (3.4.14) in seconds

The following examples show that the lexicographical term order can be "faster" than the reverse lexicographical term order and vice versa, and show that the order of variables may lead to a large difference between the total run-times.

Example 3.4.15. The (non-full) matrix

$$M_6 := \begin{pmatrix} 1 & x & 0 & 0 & 0 \\ 0 & 1 & y & 0 & 0 \\ -z & 0 & 0 & 0 & z \\ -x & 0 & 0 & 2 & x \\ y & 0 & 0 & 0 & -y \end{pmatrix}$$

results in non-trivial ideals with the reverse lexicographical term order of both variable orders, whereas the lexicographical term order only gives a result with the order of

full?	fullHDMP?	fullQP?	fullQPHDMP?
CSE	207.45	196.57	144.97

Table 3.7: Runtime for the matrix M_6 in seconds

variables $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 5$ and $j = 1, \ldots, 5$. The lexicographical term order with variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 5$ and $j = 1, \ldots, 5$ returns a "Control stack exhausted" error. For more details see Table 3.7.

Example 3.4.16. The (full) matrix

$$M_4 := \begin{pmatrix} 1 & x & 0 & 0 & 0 \\ 0 & z+2x & 0 & 0 & 1 \\ 0 & 0 & -z & 0 & z \\ y & 0 & 0 & x & x \\ 0 & 0 & -y+x & 0 & -y \end{pmatrix}$$

only results in trivial ideals with the lexicographical term order and the variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 5$ and $j = 1, \ldots, 5$. All other combinations return a "Control stack exhausted" error. In Table 3.8 the run-time is shown.

full?	fullHDMP?	fullQP?	fullQPHDMP?
362.96	CSE	CSE	CSE

Table 3.8: Runtime for the matrix M_4 in seconds

Example 3.4.17. The total run-times for the non-full matrix

$$N_3 := \begin{pmatrix} 1 & x & 0 & 0 \\ 0 & 1 & 0 & -y \\ 0 & 0 & 0 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

do not depend on the choice of the variable order and the choice of the term order. In Table 3.9 the total run-times are shown.

full?	fullHDMP?	fullQP?	fullQPHDMP?
0.23	0.19	0.22	0.18

Table 3.9: Runtime for the matrix N_3 in seconds

Example 3.4.18. The result of the full matrix

$$N_2 := \begin{pmatrix} 1 & -x & -y & 0 \\ 0 & 1 & 0 & -x \\ -z & 0 & -x & 0 \\ 0 & -z & 0 & 0 \end{pmatrix}$$

with the reverse lexicographical term order is faster than with the lexicographical term order (independent from the variable order). The variable order does not really influence the reverse lexicographical term order, but in the lexicographical term order the variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 4$ and $j = 1, \ldots, 4$ is faster than the variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 4$ and $j = 1, \ldots, 4$. In Table 3.10 the total run-times are shown.

`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
1.99	1.14	1.57	1.24

Table 3.10: Runtime for the matrix N_2 in seconds

Example 3.4.19. The full matrix

$$M_8 := \begin{pmatrix} 1 & x & 0 & 0 & 0 \\ 0 & 1 & y & 0 & 0 \\ 0 & 0 & 0 & 0 & z \\ 0 & 0 & 0 & 2 & x \\ 0 & 0 & x & 0 & -y \end{pmatrix}$$

results with the reverse lexicographical term order faster than with the lexicographical term order independently of the variable order. In this case the best choice is the reverse lexicographical term order with variable order $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 5$ and $j = 1, \ldots, 5$ to receive a result in approximately one minute. In contrast, in approximately two minutes the lexicographical term order with variable order $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ for $i = 1, \ldots, 5$ and $j = 1, \ldots, 5$ results in trivial ideal to determine the matrix as full matrix. Table 3.11 includes the exactly total run-times.

`full?`	`fullHDMP?`	`fullQP?`	`fullQPHDMP?`
77.75	71.34	121.81	65.96

Table 3.11: Runtime for the matrix M_8 in seconds

Conclusion. We were not able to find a general rule for the choice of a specific term order or variable order for minimizing the total run-time. The best choice of term order and variable order depends on the specific example. Under consideration there are several other possibilities of variable orders which may result in lower run-time.

Conclusion and Outlook

In this thesis we are able to handle (for the factorization) non-commutative polynomials, represented by a minimal admissible linear system $As = v$,

- up to rank 4 if they are *reducible* polynomials and we give explicit solution(s) of the factorization of the polynomials.
- up to rank 17 if they are *reducible* polynomials and we determine the minimal ranks of their factors.
- up to rank 12 if they are *irreducible* polynomials.

Special cases like variable disjoint factorization are not treated here. In [ARJ15] the uniqueness of the variable disjoint factorization is shown and it is computable in polynomial time.

For the factorization of a non-commutative polynomial we must find transformation matrices of the form

$$(P,\, Q) = \left(\begin{pmatrix} 1 & a_{1,\,2} & \dots & a_{1,\,n-1} & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 1 & a_{n-2,\,n-1} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & b_{2,\,3} & \dots & b_{2,\,n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & b_{n-1,\,n} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \right)$$

with entries $a_{i,j},\, b_{i,j} \in \mathbb{K}$, such that PAQ has an appropriate zero block.

In contrast to that, testing the fullness of matrices we have matrices over $\mathbb{K}$ of the form

$$P = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}, \quad Q = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & & \ddots & \vdots \\ b_{n-1,1} & b_{n-1,2} & \dots & b_{n-1,n} \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix}.$$

In the treated ideal for testing fullness we additionally impose invertibility of these two matrices by requiring $\det P = 1$ and $\det Q = 1$, whereas in the factorization our transformation matrices are invertible by construction. Therefore the generation of

the ideals in the factorization is much easier than in the test of fullness, since the latter already fails for lack of space/time resources.

Future work in testing fullness of matrices could focus on the construction of the matrices respectively reducing the "difficulties" of the determinants in the ideal.

Here the ideals are represented by Gröbner bases which depend on term and variable order. Testing the fullness of matrices we focus on the lexicographical and reverse lexicographical term order and the variable orders $\left((a_{i,j})_{i,j}, (b_{i,j})_{i,j}\right)$ respectively $\left((b_{i,j})_{i,j}, (a_{i,j})_{i,j}\right)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. There are several other possibilities of term or variable orders which may result in lower run-time. For future work here are some other possibilities for variable orders:

- Random permutations of the whole variable list
- Random permutations among $a_{i,j}$'s or $b_{i,j}$'s
- Partitioning our variables into one partition V_1 of variables being part of the Gröbner basis and another partition V_2 of the remaining variables. The variable order consists of (V_1, V_2), i.e., first the variables used in the Gröbner basis, then the remaining variables.

Bibliography

[ARJ15] V. Arvind, G. Rattan, and P. Joglekar. "On the complexity of noncommutative polynomial factorization". In: *Mathematical foundations of computer science 2015. Part II*. Vol. 9235. Lecture Notes in Comput. Sci. Springer, Heidelberg, 2015, pp. 38–49. DOI: 10.1007/978-3-662-48054-0_4. URL: http://dx.doi.org/10.1007/978-3-662-48054-0_4.

[BS15] N. R. Baeth and D. Smertnig. "Factorization theory: from commutative to noncommutative settings". In: *J. Algebra* 441 (2015), pp. 475–551. ISSN: 0021-8693. DOI: 10.1016/j.jalgebra.2015.06.007. URL: http://dx.doi.org/10.1016/j.jalgebra.2015.06.007.

[Buc65] B. Buchberger. "Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal". PhD thesis. Universität Innsbruck, Österreich, 1965.

[Car10] F. Caruso. "Factorization of Non-Commutative Polynomials". In: *ArXiv e-prints* (Feb. 2010). arXiv: 1002.3180 [cs.MS].

[CCT08] Fabrizio Caruso, Pasqualina Conti, and Carlo Traverso. "Non-commutative factorisation and GCD with applications to public key cryptography". In: *Proceedings of Differential Algebra and Related Computer Algebra*. 2008, pp. 37–39.

[Coh63] P. M. Cohn. "Noncommutative unique factorization domains". In: *Trans. Amer. Math. Soc.* 109 (1963), pp. 313–331. ISSN: 0002-9947.

[Coh72] P. M. Cohn. "Generalized rational identities". In: *Ring theory (Proc. Conf., Park City, Utah, 1971)*. Academic Press, New York, 1972, pp. 107–115.

[Coh85] P. M. Cohn. *Free rings and their relations*. Second. Vol. 19. London Mathematical Society Monographs. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], London, 1985, pp. xxii+588. ISBN: 0-12-179152-1.

[Coh95] P. M. Cohn. *Skew fields*. Vol. 57. Encyclopedia of Mathematics and its Applications. Theory of general division rings. Cambridge University Press, Cambridge, 1995, pp. xvi+500. ISBN: 0-521-43217-0. URL: http://dx.doi.org/10.1017/CBO9781139087193.

[Coh03] P. M. Cohn. *Further algebra and applications*. Springer-Verlag London, Ltd., London, 2003, pp. xii+451. ISBN: 1-85233-667-6. DOI: 10.1007/978-1-4471-0039-3. URL: http://dx.doi.org/10.1007/978-1-4471-0039-3.

[CR94] P. M. Cohn and C. Reutenauer. "A normal form in free fields". In: *Canad. J. Math.* 46.3 (1994), pp. 517–531. ISSN: 0008-414X. DOI: 10.4153/CJM-1994-027-4. URL: http://dx.doi.org/10.4153/CJM-1994-027-4.

[CR99] P. M. Cohn and C. Reutenauer. "On the construction of the free field". In: *Internat. J. Algebra Comput.* 9.3-4 (1999). Dedicated to the memory of Marcel-Paul Schützenberger, pp. 307–323. ISSN: 0218-1967. DOI: 10.1142/S0218196799000205. URL: http://dx.doi.org/10.1142/S0218196799000205.

[CLO15] D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Fourth. Undergraduate Texts in Mathematics. An introduction to computational algebraic geometry and commutative algebra. Springer, Cham, 2015, pp. xvi+646. ISBN: 978-3-319-16720-6; 978-3-319-16721-3. DOI: 10.1007/978-3-319-16721-3. URL: http://dx.doi.org/10.1007/978-3-319-16721-3.

[FR04] M. Fortin and C. Reutenauer. "Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank". In: *Sém. Lothar. Combin.* 52 (2004), Art. B52f, 12 pp. (electronic). ISSN: 1286-4889.

[Gil15] L. A. Gilch. "Symbolic Computation". In: *Vorlesungsmitschrift, TU Graz* (SS 2015).

[Jan18] B. Janko. "Factorization of non-commutative Polynomials and Testing Full Matrices". In: *Projektarbeit, TU Graz* (WS 2017/18).

[Sch17a] K. Schrempf. "Linearizing the Word Problem in (some) Free Fields". In: *ArXiv e-prints* (Jan. 2017). arXiv: 1701.03378 [math.RA].

[Sch17b] K. Schrempf. "On the Factorization of Non-Commutative Polynomials (in Free Associative Algebras)". In: *ArXiv e-prints* (June 2017). arXiv: 1706.01806 [math.RA].

[Sme15] D. Smertnig. "Factorizations of Elements in Noncommutative Rings: A Survey". In: *ArXiv e-prints* (July 2015). arXiv: 1507.07487 [math.RA].