`fdalg.spad`
Free Fields in FriCAS

Konrad Schrempf*

September 7, 2018

Abstract

The domain constructor for the field of rational numbers $\mathbb{Q}$ in FriCAS is `FRAC(INT)`. What would happen if we "replace" the ring of the integers $\mathbb{Z}$ by the *free associative algebra* $\mathbb{Q}\langle X\rangle$ (which does not have zero-devisors), that is, the non-commutative polynomials over the alphabet $X = \{x, y, z\}$ and the base field $\mathbb{Q}$? For a thourough answer one has to dig very deep into non-commutative algebra. Alghouth there could be several different *fields of fractions*, the *free* $\mathbb{Q}$-*algebra* $\mathbb{Q}\langle X\rangle$ admits a *universal field of fracions* $\mathbb{F} = \mathbb{Q}(\langle X\rangle)$ and it is possible to work with its elements in terms of *linear representations* (aka "free fractions"). The domain constructor for $\mathbb{F} = \mathbb{Q}(\langle X\rangle)$ is `FDALG(OVAR[x,y,z],FRAC(INT))` which could be imagined as "`FRAC(XDPOLY(OVAR[x,y,z],FRAC(INT)))`".

Contents

*Contact: math@versibilitas.at

Warning. This is only a very rough documentation of a highly experimental implementation which has been tested mainly for the base field `FRAC(INT)`. It serves as a technical addendum to the mathematical theory described in [Sch17b], [Sch17c], [Sch17a], [Sch18a] and [Sch18b]. For further details we refer to the book [Coh06, Chapter 7] and the work of Cohn and Reutenauer [CR94, CR99]. Additonal remarks (in German) on the implementation can be found in [Sch18c, Section B.5].

Introduction

Although the *free field* is "just" a (skew-)field —in particular, every non-zero element is invertible— it is not that easy to understand (like for example the field of the rational numbers $\mathbb{Q}$). The development of the theory took almost four decades [Coh06, Chapter 7], a lot of mathematicians contributed. For now almost another five decades free fields were hard to apply because it was not possible to work with its elements in computer algebra systems (partly except for special cases like non-commutative polynomials or rational formal power series).

This (experimental) implementation of the free field should help to explore a fascinating world which is almost inaccessible without a degree in mathematics and difficult without a specialization in (non-commutative) algebra. It should be seen like the way we learn fractions in school, long before we learn how to construct the rational numbers out of the ring of the integers ...

Each element f in the free field $\mathbb{F} = \mathbb{K}\langle\!(X)\!\rangle$ can be written in the form $f = uA^{-1}v$ with a "full" (that can be thought as a generalization of invertibility) system matrix $A = (a_{ij})$ (of some dimension n) with entries of the form $a_{ij} = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_d x_d$ for d *letters* $x_i \in X$ and *scalars* $\alpha_i \in \mathbb{K}$. The row vector u and the column vector v have scalar entries. Here usually we have $X = \{x, y, z\}$ and $\mathbb{K} = \mathbb{Q}$. The triple (u, A, v) is called *linear representation* of f. Usually we have $u = [1, 0, \ldots, 0]$. Then we call (u, A, v) an *admissible linear system* (ALS for short) and write it also as $As = v$ or $\mathcal{A} = \mathcal{A}_f = (u, A, v)$. The first component s_1 of the (unique) solution vector s is f. Admissible linear systems can be seen as a generalization of fractions, so one could call them "free fractions".

For the implementation a list of (square) matrices of size $n + 1$ is used. For $f = (x - xyx)^{-1}$ with respect to the monomials $(1, x, y)$ we have

$$\mathcal{A} = (u, A, v) = \left(\begin{bmatrix} 1 & . & . \end{bmatrix}, \begin{bmatrix} x & 1 & . \\ . & y & -1 \\ . & -1 & x \end{bmatrix}, \begin{bmatrix} . \\ . \\ 1 \end{bmatrix} \right)$$

$$\text{“}{=}\text{”} \begin{bmatrix} 0 & u \\ v & A \end{bmatrix} = \left(\left[\begin{array}{c|ccc} 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{array}\right], \left[\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right], \left[\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \right).$$

Remark. Notice that admissible linear systems (as representations of elements in the free field) are far from unique even if they are *minimal*. Therefore it is necessary to bring them to a form which simplifies the comparison of different "free fractions".

1 Initialization

For the examples here we use the following (basic) setup in FriCAS (revision 2398) [Fri18]. Only a limited number of commands is necessary for "daily" work. More information can be listed by `)show FDALG` respectively `)show FDALG` and for non-commutative polynomials in the standard `)show XDPOLY`. Furthermore the source files `fdalg.spad` and `linpen.spad` contain short descriptions of all implemented commands.

```
)compile linpen.spad
)compile fdalg.spad
ALPHABET := ['x, 'y, 'z];
OVL ==> OrderedVariableList(ALPHABET)
OFM ==> FreeMonoid(OVL)
K ==> Fraction(Integer)
XDP ==> XDPOLY(OVL, K)
FDA ==> FDALG(OVL, K)
x := 'x::OFM;
y := 'y::OFM;
z := 'z::OFM;
OF ==> OutputForm
DOn ==> enableDebugOutput
DOff ==> disableDebugOutput
AOn ==> enableAlternativeOutput
AOff ==> disableAlternativeOutput
```

```
    "()"
```

2 Matrix Pencils

The domain `FDALG` is built on the domain `LINPEN` which provides all the basic functionality for the *simultaneous* transformation of the (list of) coefficient matrices. Most of the functions are very simple.

2.1 Linear Solver

The function `blockElimination` uses *linear* systems of equations to find invertible transformation matrices for creating "upper right" blocks of zeros. (Instead of creating linear pencils manually we construct them via `FDALG`.)

```
g_11 : FDA := x+x*y*x;
addRows!(g_11, 3, 1, -1);
lmmp_11 := pencil(g_11)
```

$$\left[\begin{bmatrix} . & 1 & . & . & . \\ . & 1 & . & -1 & . \\ . & . & 1 & . & -1 \\ . & . & . & 1 & . \\ 1 & . & . & . & 1 \end{bmatrix}, \begin{bmatrix} . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & -1 & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix}, \begin{bmatrix} . & . & . & . & . \\ . & . & -1 & . & 1 \\ . & . & . & . & . \\ . & . & . & . & -1 \\ . & . & . & . & . \end{bmatrix}\right]$$

```
trns_11 := blockElimination(lmmp_11, [4], [2,3], [], [2,3,4], [5], [])
```

$$\left[\begin{bmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & . \\ . & . & . & 1 & . \\ . & . & . & . & 1 \end{bmatrix}, \begin{bmatrix} 1 & . & . & . & . \\ . & 1 & . & . & . \\ . & . & 1 & . & 1 \\ . & . & . & 1 & . \\ . & . & . & . & 1 \end{bmatrix}\right]$$

```
lmmp_12 := copy(lmmp_11);
transformRows!(lmmp_12, trns_11(1));
transformColumns!(lmmp_12, trns_11(2));
lmmp_12
```

$$\left[\begin{bmatrix} . & 1 & . & . & . \\ . & 1 & . & -1 & . \\ . & . & 1 & . & . \\ . & . & . & 1 & . \\ 1 & . & . & . & 1 \end{bmatrix}, \begin{bmatrix} . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & -1 & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix}, \begin{bmatrix} . & . & . & . & . \\ . & . & -1 & . & . \\ . & . & . & . & . \\ . & . & . & . & -1 \\ . & . & . & . & . \end{bmatrix}\right]$$

Notice that here we do not want to change the the first row or the first column. On the level of the admissible linear system the creation of an upper right block of zeros of size 2×1 corresponds to a factorization:

```
g_12 := copy(g_11)
```

$$\begin{bmatrix} 1 & -x & -1 & x \\ & 1 & -y & -1 \\ & & 1 & -x \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ 1 \end{bmatrix}; \quad \text{MR}; \quad x + x\ y\ x$$

```
addColumns!(g_12, 2, 4, 1);
g_12
```

$$\begin{bmatrix} 1 & -x & -1 & . \\ & 1 & -y & . \\ & & 1 & -x \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ 1 \end{bmatrix}; \quad \text{MR}; \quad (1 + x\ y)\ x$$

2.2 Solving Polynomial Systems

If row and column transformations "overlap", the system to solve (for the creation of zero blocks) is no longer linear. Solving systems of polynomial equations in general is very difficult, in particular if the field is not algebraically closed. To describe the approach used in `LINPEN` in details is out of scope. We just illustrate a typical application.

```
g_13 := g_11^-1;
addColumns!(g_13, 3, 2, 1);
swapRows!(g_13, 1, 2);
swapColumns!(g_13, 2, 3);
addRows!(g_13, 1, 3, 3);
lmmp_13 := pencil(g_13)
```

$$\left[\begin{bmatrix} . & 1 & . & . \\ . & 1 & -1 & -1 \\ . & . & . & -1 \\ 1 & 3 & -3 & -3 \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & 1 \\ . & . & . & . \\ . & . & . & 3 \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & 1 & . & . \\ . & . & 1 & 1 \end{bmatrix}\right]$$

```
eliminationTransformations(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4])
```

$$\left[\begin{bmatrix} 1 & . & . & . \\ . & a_1 & a_2 & . \\ . & a_3 & a_4 & . \\ . & a_5 & a_6 & 1 \end{bmatrix}, \begin{bmatrix} 1 & . & . & . \\ . & 1 & . & . \\ . & b_1 & b_2 & b_3 \\ . & b_4 & b_5 & b_6 \end{bmatrix}\right]$$

We want to create a "lower left" block of zeros of size 2×1 (with respect to the lower right 3×3 subpencil). For debugging it is possible to inspect the equations in the corresponding positions in the matrices of the pencil. The list (of equations) includes the conditions $\det P = 1$ and $\det Q = 1$ to guarantee invertibility of the tranformation matrices. To find a solution we need to compute a Gröbner basis ...

```
eliminationEquations(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4], [3,4], [2])
```

$$\left[\begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & (-a_4 - a_3)\, b_4 - a_3\, b_1 + a_3 & . & . \\ . & (-a_6 - a_5 - 3)\, b_4 + (-a_5 - 3)\, b_1 + a_5 + 3 & . & . \end{bmatrix}, \right.$$
$$\left. \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & a_3\, b_4 & . & . \\ . & (a_5 + 3)\, b_4 & . & . \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & a_4 & . & . \\ . & b_4 + b_1 + a_6 & . & . \end{bmatrix}\right]$$

```
eqns_13 := eliminationEquations(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4], _
             [3,4], [2], [], [], [], [])
```

$$\Big[-b_2\ b_6 + b_3\ b_5 + 1, \quad -a_1\ a_4 + a_2\ a_3 + 1,$$
$$(-a_6 - a_5 - 3)\ b_4 + (-a_5 - 3)\ b_1 + a_5 + 3,$$
$$(-a_4 - a_3)\ b_4 - a_3\ b_1 + a_3, \quad (a_5 + 3)\ b_4, \quad a_3\ b_4, \quad b_4 + b_1 + a_6, \quad a_4\Big]$$

```
groe_13 := eliminationGroebner(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4], _
                [3,4], [2], [], [], [], [])
```

$$[b_2\ b_6 - b_3\ b_5 - 1,\ a_4,\ a_2\ a_3 + 1,\ b_4,\ b_1 - 1,\ a_6 + 1]$$

Since —if there is a solution at all— there might be several solutions one can try to “keep” other zeros by additonal equations. In our case we should try if it is possible to create also a zero entry in row 2/column 4. Finally we create an ALS by specifying a pencil to “see” the result.

```
groe_13 := eliminationGroebner(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4], _
                [3,4], [2], [2], [4], [], [])
```

$$\left[b_6 - a_1\ a_3\ b_3,\ b_3\ b_5 - a_1\ a_3\ b_2\ b_3 + 1,\ a_4,\ a_2\ a_3 + 1,\ {a_1}^2,\ b_4,\ b_1 - 1,\ a_6 + 1\right]$$

```
sol_13 := eliminationSolve(groe_13)
```

$$\big[[b_5 = -1,\ b_3 = 1,\ b_2 = 1,\ a_3 = -1,\ a_2 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_5 = -1,\ b_3 = 1,\ b_2 = 0,\ a_3 = -1,\ a_2 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_3 = -1,\ b_5 = 1,\ b_2 = 1,\ a_3 = -1,\ a_2 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_3 = -1,\ b_5 = 1,\ b_2 = 0,\ a_3 = -1,\ a_2 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_5 = -1,\ b_3 = 1,\ b_2 = 1,\ a_2 = -1,\ a_3 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_5 = -1,\ b_3 = 1,\ b_2 = 0,\ a_2 = -1,\ a_3 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_3 = -1,\ b_5 = 1,\ b_2 = 1,\ a_2 = -1,\ a_3 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0],$$
$$[b_3 = -1,\ b_5 = 1,\ b_2 = 0,\ a_2 = -1,\ a_3 = 1,\ b_4 = 0,\ a_6 = -1,\ b_1 = 1,\ a_1 = 0,\ a_4 = 0,\ b_6 = 0]\big]$$

```
trns_13 := eliminationTransformations(lmmp_13, [2,3,4], [2,3], [3,4], [2,3,4], _
                first(sol_13))
```

$$\left[\begin{bmatrix} 1 & . & . & . \\ . & . & 1 & . \\ . & -1 & . & . \\ . & . & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & . & . & . \\ . & 1 & . & . \\ . & 1 & 1 & 1 \\ . & . & -1 & . \end{bmatrix}\right]$$

```
lmmp_14 := trns_13(1) * lmmp_13 * trns_13(2)
```

$$\left[\begin{bmatrix} . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & 1 \\ 1 & . & -1 & -3 \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & . & 1 & . \\ . & . & -3 & . \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & 1 & . & . \\ . & . & . & . \\ . & . & . & 1 \end{bmatrix}\right]$$

```
g_14 := new(lmmp_14, [1,y,x]$List(OFM))
```

$$\begin{bmatrix} x & 1 & . \\ & y & 1 \\ & -1-3\,y & -3+x \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 1 \end{bmatrix};\ ??;\ \ (-x)^{-1}\,(-1-x\,y)^{-1}$$

Notice that the latter is for experimental use only because no check on the *fullness* of the system matrix is done and it should be clear that a priori nothing is known on minimality or refinement of the ALS.

Remark. Please be aware that before changing the base field `K` to a more general *commutative* field (`FRAC(POLY(INT))`, etc.) the solver `eliminationSolve` might need some adaption and has to be tested thouroghly!

3 Free Division Algebra

Beside the *admissible linear system* (which can hided by `DOff`) and a *rational expression* (which can be hided by `AOff`) an “M” (respectively “R”) is shown if the system is *minimal* (respectively *refined*). Providing a “readable” output is rather complicated and it is necessary to analyse the block structure, try to find factors and summands, invert them to check if it is a polynomial (which is easy to print), etc. So if there is some unexpected error, the alternative output should be deactivated (or supressed by appending a semicolon after the command).

Since solving polynomial systems of equations is computationally very expensive, there are special functions that just analyse the block structure of the system matrix (like `factors`) or use linear techniques only (like `summands`). These functions are mainly for preparing a readable output and *not* for judging something like *irreducibility* (of an element).

Remark. Notice that there is nothing special with a zero summand if an ALS is not minimal. If a “subsystem” $\mathcal{A}_g$ cannot be written in a nice form the corresponding term g is printed in parantheses in the form $r\langle\operatorname{rank} g\rangle$ if the rank is known or $d\langle\dim \mathcal{A}_g\rangle$ otherwise.

3.1 Polynomials

The best way to get aquainted with the free field is to start with polynomials, since one can work in parallel with the domain `XDPOLY`.

```
q_01 : XDP := 1 - x*y
```

$$1 - x\ y$$

```
q_02 : XDP := q_01 + 2*q_01
```

$$3 - 3\ x\ y$$

```
q_03 := q_02::FDA
```

$$\begin{bmatrix} 1 & -x & -1 \\ & 1 & y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 3 \end{bmatrix}; \ \text{MR}; \ \ 3 - 3\ x\ y$$

```
q_04 := q_03::XDP
```

$$3 - 3\ x\ y$$

```
p_01 : FDA := 1 - x*y;
p_02 := p_01 + 2*p_01
```

$$\begin{bmatrix} 1 & -\frac{3}{2}\ x & \frac{3}{2} \\ & 1 & -y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -2 \end{bmatrix}; \ \text{MR}; \ \ 3 - 3\ x\ y$$

```
rank(p_02)
```

$$3$$

```
p_03 := addALS(p_01, p_01)
```

$$\begin{bmatrix} 1 & -x & 1 & -1 & . & . \\ & 1 & -y & . & . & . \\ & & 1 & . & . & . \\ & & & 1 & -x & 1 \\ & & & & 1 & -y \\ & & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -1 \\ . \\ . \\ -1 \end{bmatrix}; \ \text{?R}; \ \ (1 - x\ y) + 1 - x\ y$$

```
dimension(p_03)
```

$$6$$

```
addRows!(p_03, 6, 3, -1);
addColumns!(p_03, 3, 6, 1);
p_03
```

$$\begin{bmatrix} 1 & -x & 1 & -1 & . & 1 \\ & 1 & -y & . & . & -y \\ & & 1 & . & . & . \\ & & & 1 & -x & 1 \\ & & & & 1 & -y \\ & & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ . \\ -1 \end{bmatrix}; \ ?\mathrm{R}; \ (0) + 2 - 2\ x\ y$$

```
p_04 := removeRowsColumns(p_03, [3], [3])
```

$$\begin{bmatrix} 1 & -x & -1 & . & 1 \\ & 1 & . & . & -y \\ & & 1 & -x & 1 \\ & & & 1 & -y \\ & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ -1 \end{bmatrix}; \ ?\mathrm{R}; \ (0) + 2 - 2\ x\ y$$

```
addRows!(p_04, 3, 1, 1);
p_05 := removeRowsColumns(p_04, [3], [3])
```

$$\begin{bmatrix} 1 & -x & -x & 2 \\ & 1 & . & -y \\ & & 1 & -y \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ -1 \end{bmatrix}; \ ?\mathrm{R}; \ 2 - 2\ x\ y$$

```
p_06 := minimize(p_05)
```

$$\begin{bmatrix} 1 & -2\ x & 2 \\ & 1 & -y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -1 \end{bmatrix}; \ \mathrm{MR}; \ 2 - 2\ x\ y$$

```
rank(p_06)
```

3

```
p_11 : FDA := x*y*x*y
```

$$\begin{bmatrix} 1 & -x & . & . & . \\ & 1 & -y & . & . \\ & & 1 & -x & . \\ & & & 1 & -y \\ & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ 1 \end{bmatrix}; \ \mathrm{MR}; \ x\ y\ x\ y$$

```
p_12 := copy(p_11);
variables(p_12)
```

$$[1,\ y,\ x]$$

```
appendSupport!(p_12, [z]);
variables(p_12)
```

$$[1,\ y,\ x,\ z]$$

```
p_12(1,5) := -z::XDP;
p_12 := minimize(p_12)
```

$$\begin{bmatrix} 1 & -x & . & . & -z \\ & 1 & -y & . & . \\ & & 1 & -x & . \\ & & & 1 & -y \\ & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ 1 \end{bmatrix};\ \text{MR};\ \ z + x\ y\ x\ y$$

```
p_13 := 3::FDA
```

$$3$$

```
admissibleLinearSystem(p_13)
```

$$\begin{bmatrix} 1 \end{bmatrix} s = \begin{bmatrix} 3 \end{bmatrix}$$

```
p_14 : FDA := x*y+z;
representation(p_14)
```

$$\left[\begin{bmatrix} 1 & . & . \end{bmatrix}, \begin{bmatrix} 1 & -x & -z \\ . & 1 & -y \\ . & . & 1 \end{bmatrix}, \begin{bmatrix} . \\ . \\ 1 \end{bmatrix} \right]$$

```
pencil(p_14)
```

$$\left[\begin{bmatrix} . & 1 & . & . \\ . & 1 & . & . \\ . & . & 1 & . \\ 1 & . & . & 1 \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & -1 \\ . & . & . & . \\ . & . & . & . \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ . & . & . & -1 \\ . & . & . & . \end{bmatrix}, \begin{bmatrix} . & . & . & . \\ . & . & -1 & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \right]$$

```
multiplyRow!(p_14, 3, 7);
p_14
```

$$\begin{bmatrix} 1 & -x & -z \\ & 1 & -y \\ & & 7 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 7 \end{bmatrix}; \text{ MR; } \ z + x\ y$$

```
normalize!(p_14);
p_14
```

$$\begin{bmatrix} 1 & -x & -z \\ & 1 & -y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 1 \end{bmatrix}; \text{ MR; } \ z + x\ y$$

Inverting (non-zero) polynomials is easy. But notice that while (the regular) f_{07} still has a representation as formal power series, f_{09} does not admit such a representation any more ...

```
f_07 := p_02^-1
```

$$\begin{bmatrix} y & \frac{1}{2} \\ 2 & x \end{bmatrix} s = \begin{bmatrix} . \\ \frac{2}{3} \end{bmatrix}; \text{ MR; } \ (3 - 3\ x\ y)^{-1}$$

```
f_08 := f_07^-1
```

$$\begin{bmatrix} 1 & -\frac{3}{2}\ x & -3 \\ & 1 & 2\ y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 1 \end{bmatrix}; \text{ MR; } \ 3 - 3\ x\ y$$

```
f_09 := p_15^-1
```

$$\begin{bmatrix} x & -1 & . \\ 1 & y & -1 \\ & & x \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 1 \end{bmatrix}; \text{ MR; } \ (1 + y\ x)^{-1}\ x^{-1}$$

```
f_10 : FDA := 2/7
```

$$\frac{2}{7}$$

```
f_10^-1
```

$$\frac{7}{2}$$

3.2 Factorization

Factorization in free associative algebras can be generalized to their respective free field in terms of *minimal* linear representations [Sch17a]. However, since the necessary functionality is not yet fully implemented, we restrict here to polynomials only.

Notice that $\mathbb{K}\langle X\rangle$ is a *similarity-unique factorization domain* (similarity UFD), so uniqueness is only up to *similarity*, for example $x - xyx = x(1-yx) = (1-xy)x$.

Irreducibility of p_{12} can be checked easily by testing if it has a non-trivial (left) factor, that is, one of rank 2, 3 or 4. In the case of reducibility the returned list would have two entries. Alternatively `factor(p_12)` can be used.

```
factorize(p_12, 2)
```

$$\left[\left[\begin{array}{ccccc} 1 & -x & . & . & -z \\ & 1 & -y & . & . \\ & & 1 & -x & . \\ & & & 1 & -y \\ & & & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ . \\ . \\ 1 \end{array}\right]; \ \mathrm{MR}; \ z + x\ y\ x\ y\right]$$

```
factorize(p_12, 3)
```

$$\left[\left[\begin{array}{ccccc} 1 & -x & . & . & -z \\ & 1 & -y & . & . \\ & & 1 & -x & . \\ & & & 1 & -y \\ & & & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ . \\ . \\ 1 \end{array}\right]; \ \mathrm{MR}; \ z + x\ y\ x\ y\right]$$

```
factorize(p_12, 4)
```

$$\left[\left[\begin{array}{ccccc} 1 & -x & . & . & -z \\ & 1 & -y & . & . \\ & & 1 & -x & . \\ & & & 1 & -y \\ & & & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ . \\ . \\ 1 \end{array}\right]; \ \mathrm{MR}; \ z + x\ y\ x\ y\right]$$

```
p_15 : FDA := x+x*y*x
```

$$\left[\begin{array}{cccc} 1 & -x & . & . \\ & 1 & -y & -1 \\ & & 1 & -x \\ & & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ . \\ 1 \end{array}\right]; \ \mathrm{MR}; \ x\ (1 + y\ x)$$

```
factorize(p_15, 3)
```

$$\left[\left[\begin{array}{ccc} 1 & -x & -1 \\ & 1 & -y \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ 1 \end{array}\right];\ \text{MR};\ 1+x\,y,\ \left[\begin{array}{cc} 1 & -x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ 1 \end{array}\right];\ \text{MR};\ x\right]$$

```
factorize(p_15, 2)
```

$$\left[\left[\begin{array}{cc} 1 & -x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ 1 \end{array}\right];\ \text{MR};\ x,\ \left[\begin{array}{ccc} 1 & -y & -1 \\ & 1 & -x \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ 1 \end{array}\right];\ \text{MR};\ 1+y\,x\right]$$

Some of the following examples are not that trivial as they look at a first glance. Notice in particular how the output (as rational expression) changes with respect to the (upper right) zeros in the system matrix. And due to non-commutativity, we have $p_{33} \neq p_{34}$.

```
p_31 : FDA := (x-1)*(x+1)
```

$$\left[\begin{array}{ccc} 1 & 1-x & . \\ & 1 & -1-x \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ 1 \end{array}\right];\ \text{MR};\ (-1+x)\,(1+x)$$

```
factors(p_31)
```

$$\left[\left[\begin{array}{cc} 1 & 1-x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ 1 \end{array}\right];\ \text{MR};\ -1+x,\ \left[\begin{array}{cc} 1 & -1-x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ 1 \end{array}\right];\ \text{MR};\ 1+x\right]$$

```
p_32 : FDA := x^2 - 1
```

$$\left[\begin{array}{ccc} 1 & -x & -1 \\ & 1 & x \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ -1 \end{array}\right];\ \text{MR};\ -1+x^2$$

```
factors(p_32)
```

$$\left[\left[\begin{array}{ccc} 1 & -x & -1 \\ & 1 & x \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ -1 \end{array}\right];\ \text{MR};\ -1+x^2\right]$$

```
factorize(p_32, 2)
```

$$\left[\left[\begin{array}{cc} 1 & -1-x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ 1 \end{array}\right];\ \text{MR};\ 1+x,\ \left[\begin{array}{cc} 1 & -1+x \\ & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ -1 \end{array}\right];\ \text{MR};\ -1+x\right]$$

```
addRows!(p_32, 2, 1, 1);
addColumns!(p_32, 2, 3, 1);
p_32
```

$$\left[\begin{array}{ccc} 1 & 1-x & . \\ & 1 & 1+x \\ & & 1 \end{array}\right] s = \left[\begin{array}{c} . \\ . \\ -1 \end{array}\right];\ \text{MR};\ (-1+x)\,(1+x)$$

```
p_33 : FDA := (x-y)*(x+y)
```

$$\begin{bmatrix} 1 & y-x & . \\ & 1 & -y-x \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ 1 \end{bmatrix}; \text{ MR; } (-y+x)\,(y+x)$$

```
p_34 : FDA := x^2 - y^2
```

$$\begin{bmatrix} 1 & -x & -y & . \\ & 1 & . & x \\ & & 1 & -y \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ -1 \end{bmatrix}; \text{ MR; } -y^2+x^2$$

```
p_35 := p_33 - p_34
```

$$\begin{bmatrix} 1 & -y & -x & . \\ & 1 & . & x \\ & & 1 & -y \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ 1 \end{bmatrix}; \text{ MR; } -y\;x+x\;y$$

```
p_41 : FDA := y^2 - 1
```

$$\begin{bmatrix} 1 & -y & -1 \\ & 1 & y \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -1 \end{bmatrix}; \text{ MR; } -1+y^2$$

```
fct_41 := factor(p_41)
```

$$\left[\begin{bmatrix} 1 & -1-y \\ & 1 \end{bmatrix} s = \begin{bmatrix} . \\ 1 \end{bmatrix}; \text{ MR; } 1+y, \begin{bmatrix} 1 & -1+y \\ & 1 \end{bmatrix} s = \begin{bmatrix} . \\ -1 \end{bmatrix}; \text{ MR; } -1+y\right]$$

```
reduce(*, fct_41) - p_41
```

$$0$$

```
p_42 : FDA := (1-x*y)*(2+y*x)*(3-y*z)*(2-z*y)*(1-x*z)*(3+z*x)*x;
AOff(p_42);
p_42
```

$$\begin{bmatrix}
1 & -x & -1 & . & . & . & . & . & . & . & . & . & . & . \\
 & 1 & y & . & . & . & . & . & . & . & . & . & . & . \\
 & & 1 & -y & -2 & . & . & . & . & . & . & . & . & . \\
 & & & 1 & -x & . & . & . & . & . & . & . & . & . \\
 & & & & 1 & -y & -3 & . & . & . & . & . & . & . \\
 & & & & & 1 & z & . & . & . & . & . & . & . \\
 & & & & & & 1 & -z & -2 & . & . & . & . & . \\
 & & & & & & & 1 & y & . & . & . & . & . \\
 & & & & & & & & 1 & -x & -1 & . & . & . \\
 & & & & & & & & & 1 & z & . & . & . \\
 & & & & & & & & & & 1 & -z & -3 & . \\
 & & & & & & & & & & & 1 & -x & . \\
 & & & & & & & & & & & & 1 & -x \\
 & & & & & & & & & & & & & 1
\end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ 1 \end{bmatrix}; \mathrm{MR}$$

```
p_43 := copy(p_42);
addRows!(p_43, 13, 1, 2);
addRows!(p_43, 8, 3, 1);
addColumns!(p_43, 4, 10, 3);
addColumns!(p_43, 6, 12, -1);
DOff(p_43);
p_43
```

$$\begin{aligned}
&36\ x - 18\ z\ y\ x + 12\ z\ x^2 - 12\ y\ z\ x + 18\ y\ x^2 - 36\ x\ z\ x - 36\ x\ y\ x - 6\ z\ y\ z\ x^2 \\
&\quad + 18\ z\ y\ x\ z\ x + 6\ y\ z^2\ y\ x - 4\ y\ z^2\ x^2 + 12\ y\ z\ x\ z\ x - 9\ y\ x\ z\ y\ x + 6\ y\ x\ z\ x^2 \\
&\quad - 6\ y\ x\ y\ z\ x - 18\ y\ x^2\ z\ x - 12\ x\ z^2\ x^2 + 18\ x\ y\ z\ y\ x - 12\ x\ y\ z\ x^2 \\
&\quad + 12\ x\ y^2\ z\ x - 18\ x\ y^2\ x^2 + 36\ x\ y\ x\ z\ x + 6\ z\ y\ x\ z^2\ x^2 + 2\ y\ z^2\ y\ z\ x^2 \\
&\quad - 6\ y\ z^2\ y\ x\ z\ x + 4\ y\ z\ x\ z^2\ x^2 - 3\ y\ x\ z\ y\ z\ x^2 + 9\ y\ x\ z\ y\ x\ z\ x \\
&\quad + 3\ y\ x\ y\ z^2\ y\ x - 2\ y\ x\ y\ z^2\ x^2 + 6\ y\ x\ y\ z\ x\ z\ x - 6\ y\ x^2\ z^2\ x^2 + 6\ x\ y\ z\ y\ z\ x^2 \\
&\quad - 18\ x\ y\ z\ y\ x\ z\ x - 6\ x\ y^2\ z^2\ y\ x + 4\ x\ y^2\ z^2\ x^2 - 12\ x\ y^2\ z\ x\ z\ x + 9\ x\ y^2\ x\ z\ y\ x \\
&\quad - 6\ x\ y^2\ x\ z\ x^2 + 6\ x\ y^2\ x\ y\ z\ x + 18\ x\ y^2\ x^2\ z\ x + 12\ x\ y\ x\ z^2\ x^2 \\
&\quad - 2\ y\ z^2\ y\ x\ z^2\ x^2 + 3\ y\ x\ z\ y\ x\ z^2\ x^2 + y\ x\ y\ z^2\ y\ z\ x^2 - 3\ y\ x\ y\ z^2\ y\ x\ z\ x \\
&\quad + 2\ y\ x\ y\ z\ x\ z^2\ x^2 - 6\ x\ y\ z\ y\ x\ z^2\ x^2 - 2\ x\ y^2\ z^2\ y\ z\ x^2 + 6\ x\ y^2\ z^2\ y\ x\ z\ x \\
&\quad - 4\ x\ y^2\ z\ x\ z^2\ x^2 + 3\ x\ y^2\ x\ z\ y\ z\ x^2 - 9\ x\ y^2\ x\ z\ y\ x\ z\ x - 3\ x\ y^2\ x\ y\ z^2\ y\ x \\
&\quad + 2\ x\ y^2\ x\ y\ z^2\ x^2 - 6\ x\ y^2\ x\ y\ z\ x\ z\ x + 6\ x\ y^2\ x^2\ z^2\ x^2 - y\ x\ y\ z^2\ y\ x\ z^2\ x^2 \\
&\quad + 2\ x\ y^2\ z^2\ y\ x\ z^2\ x^2 - 3\ x\ y^2\ x\ z\ y\ x\ z^2\ x^2 - x\ y^2\ x\ y\ z^2\ y\ z\ x^2 \\
&\quad + 3\ x\ y^2\ x\ y\ z^2\ y\ x\ z\ x - 2\ x\ y^2\ x\ y\ z\ x\ z^2\ x^2 + x\ y^2\ x\ y\ z^2\ y\ x\ z^2\ x^2
\end{aligned}$$

```
fct_43 := factor(p_43);
map(DOff, fct_43);
fct_43
```

$$[1 - x\ y,\ 2 + y\ x,\ 3 - y\ z,\ 2 - z\ y,\ 1 - x\ z,\ 3 + z\ x,\ x]$$

```
reduce(*, fct_43) - p_42
```

$$0$$

3.3 Hua's Identity

How to proof Hua's identity $x - \big(x^{-1} + (y^{-1} - x)^{-1}\big)^{-1} = xyx$ step by step is explained in detail in [Sch18a, Example 5.1]. In `FDALG` it is just one line ...

```
f_21 : FDA := x - (x^-1 + (y^-1 - x)^-1)^-1 - x*y*x
```

$$0$$

```
f_22 : FDA := y^-1 - x
```

$$\begin{bmatrix} y & -y & 1 \\ & 1 & -x \\ & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -1 \end{bmatrix}; \quad \text{MR}; \quad y^{-1} - x$$

```
f_23 : FDA := x^-1 + f_22^-1
```

$$\begin{bmatrix} x & . & 1 \\ & 1 & y \\ & x & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ -1 \end{bmatrix}; \quad \text{MR}; \quad x^{-1} + (r2)$$

```
f_24 : FDA := x - f_23^-1
```

$$\begin{bmatrix} 1 & -x & 1 & . \\ & 1 & -y & 1 \\ & & 1 & x \\ & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ -1 \end{bmatrix}; \quad \text{MR}; \quad x\ y\ x$$

3.4 Remarks

To understand what is going on in the background, the following commands are available to construct admissible linear systems for the sum, the product and the inverse: `addALS`, `multiplyALS` and `invertALS`. See [Sch17b, Proposition 1.13] or [Sch18a, Proposition 2.2]. Then the corresponding minimization steps can be done manually. For polynomials this is explained in [Sch17c, Section 2.2]. For the standard inverse [Sch17b, Proposition 4.2] there is `invertSTD`, and for the minimal inverse [Sch17b, Theorem 4.20] there is `invertMIN`. The latter is used very much internally and since refinement of pivot blocks needs Groebner bases techniques it tries to create a "fine" upper right block structure using *linear* techniques before "inverting" the element. To keep a "readable" admissible linear system some "normalization" is done. For details see the implementation.

3.5 Outlook

The following example should only illustrate that there is still a lot to do. Checking for refinement of pivot blocks of size greater than 4 (or 5 if there is only one pivot block) could take quite a long time. If a polynomial is inverted, irreducibility should be checked instead because the factorization transformations are invertible by definition. In this case a flag "irreducible" could be used to set the flag "refined" in the inverse.

Since usually the coefficient matrices in the pencil are rather sparse one could use a representation for non-zero entries only, that is, `SparseLinearMultivariateMatrixPencil`. This would simplify analyzing the block structure.

Working over a more general *commutative* base field than the rational numbers $\mathbb{Q}$ is not an issue with respect to the mathematical theory. The difficulty lies in the computation of a solution over non-algebraically closed fields. The special solver in `LINPEN` might have to be adapted and tested thouroughly!

```
f_51 : FDA := x*y*z + z*y*x
```

$$\begin{bmatrix} 1 & -x & . & -z & . & . \\ & 1 & -y & . & . & . \\ & & 1 & . & . & -z \\ & & & 1 & -y & . \\ & & & & 1 & -x \\ & & & & & 1 \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ . \\ 1 \end{bmatrix}; \quad \text{MR}; \quad z\ y\ x + x\ y\ z$$

```
f_52 := f_51^-1
```

$$\begin{bmatrix} x & -1 & . & . & . \\ . & y & -1 & . & . \\ z & . & . & -1 & . \\ . & . & . & y & -1 \\ . & . & z & . & x \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ 1 \end{bmatrix}; \quad \text{MR}; \quad (z\ y\ x + x\ y\ z)^{-1}$$

```
f_53 := invertMIN(f_51)
```

$$\begin{bmatrix} x & -1 & . & . & . \\ . & y & -1 & . & . \\ z & . & . & -1 & . \\ . & . & . & y & -1 \\ . & . & z & . & x \end{bmatrix} s = \begin{bmatrix} . \\ . \\ . \\ . \\ 1 \end{bmatrix}; \quad \text{M?}; \quad (z\ y\ x + x\ y\ z)^{-1}$$

Acknowledgement

I am very grateful to the lively discussions with Raymond Rogers and Bill Page which motivated me to bring my highly experimental implementation to a (hopefully) usable form.

References

[Coh06] P. M. Cohn. *Free ideal rings and localization in general rings*, volume 3 of *New Mathematical Monographs*. Cambridge University Press, Cambridge, 2006.

[CR94] P. M. Cohn and C. Reutenauer. A normal form in free fields. *Canad. J. Math.*, 46(3):517–531, 1994.

[CR99] P. M. Cohn and C. Reutenauer. On the construction of the free field. *Internat. J. Algebra Comput.*, 9(3-4):307–323, 1999. Dedicated to the memory of Marcel-Paul Schützenberger.

[Fri18] FriCAS *Computer Algebra System*, 2018. W. Hebisch, `http://axiom-wiki.newsynthesis.org/FrontPage`, `svn co svn://svn.code.sf.net/p/fricas/code/trunk fricas`.

[Sch17a] K. Schrempf. A Factorization Theory for some Free Fields. *ArXiv e-prints*, December 2017.

[Sch17b] K. Schrempf. Linearizing the word problem in (some) free fields. *ArXiv e-prints*, January 2017.

[Sch17c] K. Schrempf. On the factorization of non-commutative polynomials (in free associative algebras). *ArXiv e-prints*, June 2017.

[Sch18a] K. Schrempf. A Standard Form in (some) Free Fields: How to construct Minimal Linear Representations. *ArXiv e-prints*, March 2018.

[Sch18b] K. Schrempf. Free Fractions: An Invitation to (applied) Free Fields. *In preparation*, 1:1–30, September 2018.

[Sch18c] K. Schrempf. *Über die Konstruktion minimaler linearer Darstellungen von Elementen des freien Schiefkörpers (freier assoziativer Algebren), alias: "Das Rechnen mit Freien Brüchen"*. Dissertation, Universität Wien, 2018.