Computations over the free algebra

Grischa Studzinski

RWTH Aachen
Lehrstuhl D für Mathematik

15. Oktober 2009

freeqa.lib

Goal

Compute the $\mathbb{K}$-dimension and a $\mathbb{K}$-basis of a factor algebra of the free associative algebra

freeqa.lib

Goal

Compute the $\mathbb{K}$-dimension and a $\mathbb{K}$-basis of a factor algebra of the free associative algebra

The free algebra is realized as a letterplace-ring, a (finite) Gröbner basis of an ideal is given

Overview of the Procedures

lpDimCheck(G)

- Check, if the factor algebra is of finite dimension

Overview of the Procedures

lpDimCheck(G)

· Check, if the factor algebra is of finite dimension

Using the Ufnarovskij graph this procedure returns yes, if the dimension of the factor algebra is infinite and no otherwise.

There is no need to compute the dimension or a basis here.

Overview of the Procedures

lpSickle(G)

· Computation of a monomial $\mathbb{K}$-basis by building up a tree of the basis elements.

Overview of the Procedures

lpSickle(G)

· Computation of a monomial $\mathbb{K}$-basis by building up a tree of the basis elements.

lpKDim(G)

· Computation of the $\mathbb{K}$-dimension by counting normal words.

Overview of the Procedures

lpSickle(G)

· Computation of a monomial $\mathbb{K}$-basis by building up a tree of the basis elements.

lpKDim(G)

· Computation of the $\mathbb{K}$-dimension by counting normal words.

lpHilbert(G)

· Computation of the Hilbert-series. This is done by counting normal words in a fixed degree, thus giving us the corresponding coefficient of the series.

Overview of the Procedures

lpSickle(G)

· Computation of a monomial $\mathbb{K}$-basis by building up a tree of the basis elements.

lpKDim(G)

· Computation of the $\mathbb{K}$-dimension by counting normal words.

lpHilbert(G)

· Computation of the Hilbert-series. This is done by counting normal words in a fixed degree, thus giving us the corresponding coefficient of the series.

For all procedures one may add a degree-bound.

Overview of the Procedures

lpSickle(G)

· Computation of a monomial $\mathbb{K}$-basis by building up a tree of the basis elements.

lpKDim(G)

· Computation of the $\mathbb{K}$-dimension by counting normal words.

lpHilbert(G)

· Computation of the Hilbert-series. This is done by counting normal words in a fixed degree, thus giving us the corresponding coefficient of the series.

For all procedures one may add a degree-bound.
Procedures may be combined to save computation time.

The Structure

Only the leading monomials are needed for these computations

The Structure

Only the leading monomials are needed for these computations

The leading monomials depend on the chosen ordering, but afterward the order is not needed

The Structure

Only the leading monomials are needed for these computations

The leading monomials depend on the chosen ordering, but afterward the order is not needed

The monomials are stored as integer vectors, where every variable corresponds to a number

Example

$$x \mapsto 1 \qquad y \mapsto 2 \qquad z \mapsto 3$$

Example

$$x \mapsto 1 \qquad y \mapsto 2 \qquad z \mapsto 3$$

$$xyxzx \sim x(1)y(2)x(3)z(4)x(5) \mapsto (1,2,1,3,1)$$

The Structure

The user can use the letterplace-polynomials and does not need the integer vectors. There are procedures to switch between these two forms and all the main procedures can be called with a ideal of letterplace polynomials or a list of integer vectors.

The Structure

The user can use the letterplace-polynomials and does not need the integer vectors. There are procedures to switch between these two forms and all the main procedures can be called with a ideal of letterplace polynomials or a list of integer vectors.

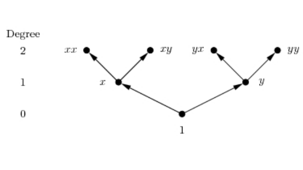
Also the procedure **sickle** allows to invoke all main procedures at once or partially. At the moment this is only possible if the input is an letterplace-ideal.

The Basis-Tree

The basis is build degree-wise: Starting in 1 we get all monomials with total degree 1 by right multiplication with all variables. All vertices representing normal words will be extended by right multiplication again. This procedure will give us a monomial basis.

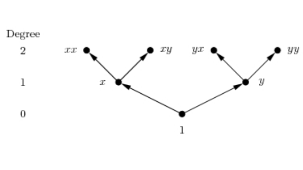
Example

A basis for $\mathbb{K}\langle x, y\rangle$ up to degree 2



Example

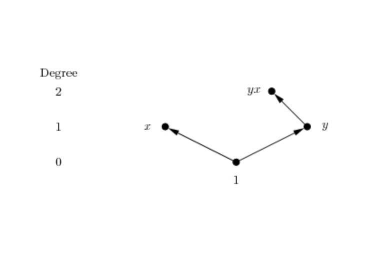
A basis for $\mathbb{K}\langle x, y \rangle$ up to degree 2



Note that every vertex has a unique path leading to it!
This is a difference to the commutative case.

Example

A basis for $\mathbb{K}\langle x, y\rangle / \langle xx, yy, xy\rangle$



Mistletoes

Observation

To each maximal vertex there is a unique path leading from 1 to this vertex and every vertex can be extended to a maximal one. Hence the whole tree is completely described by those vertices.

Mistletoes

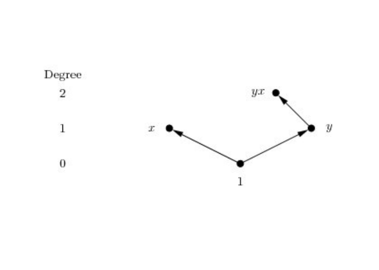
Observation

To each maximal vertex there is a unique path leading from 1 to this vertex and every vertex can be extended to a maximal one. Hence the whole tree is completely described by those vertices.

They are called **mistletoes**.

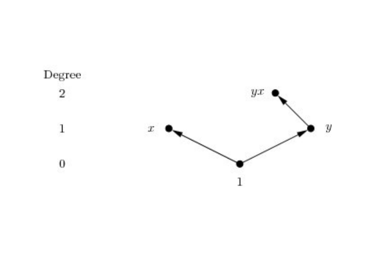
Mistletoes

A basis for $\mathbb{K}\langle x, y\rangle / \langle xx, yy, xy\rangle$



Mistletoes

A basis for $\mathbb{K}\langle x, y\rangle / \langle xx, yy, xy\rangle$



The mistletoes are $\{x, yx\}$.

Mistletoes

Mistletoes provide us with an effective way to store bases.

Mistletoes

Mistletoes provide us with an effective way to store bases.

It is easy to get the $\mathbb{K}$-basis and$\backslash$or the $\mathbb{K}$-dimension out of the mistletoes: If ordered lexicographically one only needs to compare the mistletoes that are side by side. This is due to the fact that the lexicographically order is the natural order for the basis-tree.

Example

The Symmetric Group Algebra S_3

$\mathbb{K}\langle x, y \rangle$, $\boldsymbol{G} = \{x^2 - 1, y^2 - 1, xyx - yxy\}$, $x > y$

Example

The Symmetric Group Algebra S_3

$\mathbb{K}\langle x, y\rangle,\ \boldsymbol{G} = \{x^2 - 1, y^2 - 1, xyx - yxy\},\ x > y$

Mistletoes

$M = \{xy, yxy\}$

Example

The Symmetric Group Algebra S_3

$\mathbb{K}\langle x, y\rangle$, $\boldsymbol{G} = \{x^2 - 1, y^2 - 1, xyx - yxy\}$, $x > y$

Mistletoes

$M = \{xy, yxy\}$
Since the greatest common left subword is 1 we have
$\dim_{\mathbb{K}}(\mathbb{K}\langle x, y\rangle / \langle \boldsymbol{G}\rangle) = 1 + \deg(xy) + \deg(yxy) = 6.$

Example

The Symmetric Group Algebra S_3

$\mathbb{K}\langle x, y\rangle,\ \boldsymbol{G} = \{x^2 - 1, y^2 - 1, xyx - yxy\},\ x > y$

Mistletoes

$M = \{xy, yxy\}$
Since the greatest common left subword is 1 we have
$\dim_{\mathbb{K}}(\mathbb{K}\langle x, y\rangle / \langle \boldsymbol{G}\rangle) = 1 + \deg(xy) + \deg(yxy) = 6.$

Basis

$B = \{1, x, y, xy, yx, yxy\}$

Possible Improvements

Improve the look-ups in the Gröbner basis:

- At the moment a list of intmats is searched for a common subword.
- The usage of a pointer would be very helpful to exploit the tree-structure.
- Comparison to analogues commutative procedures would be helpful.

The Future

By including the procedure **reduce** we can perform arithmetic operations in a factor algebra. So the user should be able to set it as a basering.

The Future

By including the procedure **reduce** we can perform arithmetic operations in a factor algebra. So the user should be able to set it as a basering.

We want to include integer rings as coefficient rings in the letterplace-structure, for example $\mathbb{Z}[a, b, c]$, or rings in general. This can be applied to cyclotomic Hecke-algebras to prove certain conjectures.

The Future

By including the procedure **reduce** we can perform arithmetic operations in a factor algebra. So the user should be able to set it as a basering.

We want to include integer rings as coefficient rings in the letterplace-structure, for example $\mathbb{Z}[a, b, c]$, or rings in general. This can be applied to cyclotomic Hecke-algebras to prove certain conjectures.

Computations for the free algebra: Sums, divisions, intersections of ideals, Gröbner bases of left-ideals in the factor algebra, computation of syzygies, resolution of modules.

The Future

By including the procedure **reduce** we can perform arithmetic operations in a factor algebra. So the user should be able to set it as a basering.

We want to include integer rings as coefficient rings in the letterplace-structure, for example $\mathbb{Z}[a, b, c]$, or rings in general. This can be applied to cyclotomic Hecke-algebras to prove certain conjectures.

Computations for the free algebra: Sums, divisions, intersections of ideals, Gröbner bases of left-ideals in the factor algebra, computation of syzygies, resolution of modules.

Also the codimension of one ideal with respect to another is of interest, if it is finite.

The Future

By including the procedure **reduce** we can perform arithmetic operations in a factor algebra. So the user should be able to set it as a basering.

We want to include integer rings as coefficient rings in the letterplace-structure, for example $\mathbb{Z}[a, b, c]$, or rings in general. This can be applied to cyclotomic Hecke-algebras to prove certain conjectures.

Computations for the free algebra: Sums, divisions, intersections of ideals, Gröbner bases of left-ideals in the factor algebra, computation of syzygies, resolution of modules.

Also the codimension of one ideal with respect to another is of interest, if it is finite.

Improve the letterplace structure to an effective, fast and user-friendly subsystem of Singular