

GradedRingForHomalg

Endow Commutative Rings with an Abelian Grading

Version 2014.02.04

February 2014

Mohamed Barakat

Sebastian Gutsche

Markus Lange-Hegermann

This manual is best viewed as an HTML document. The latest version is
available ONLINE at:

<http://homalg.math.rwth-aachen.de/~markus/GradedRingForHomalg/chap0.html>

An OFFLINE version should be included in the documentation subfolder
of the package.

Mohamed Barakat

Email: barakat@mathematik.uni-kl.de

Homepage: <http://www.mathematik.uni-kl.de/~barakat/>

Address: Department of Mathematics,
University of Kaiserslautern,
67653 Kaiserslautern,
Germany

Sebastian Gutsche

Email: sebastian.gutsche@rwth-aachen.de

Homepage: <http://wwwb.math.rwth-aachen.de/~gutsche/>

Address: Lehrstuhl B für Mathematik, RWTH Aachen, Templer-
graben 64, 52056 Aachen, Germany

Markus Lange-Hegermann

Email: markus.lange.hegermann@rwth-aachen.de

Homepage: <http://wwwb.math.rwth-aachen.de/~markus>

Address: Lehrstuhl B für Mathematik, RWTH Aachen, Templer-
graben 64, 52056 Aachen, Germany

Copyright

© 2008-2014 by Mohamed Barakat, Sebastian Gutsche, and Markus Lange-Hegermann

This package may be distributed under the terms and conditions of the GNU Public License Version 2.

Acknowledgements

We would like to thank the Aachen forest for being such a nice place for jogging.

Contents

1	Introduction	4
1.1	What is the Role of the GradedRingForHomalg Package in the homalg Project? . .	4
1.2	Functionality	4
1.3	The Math Behind This Package	4
2	Installation of the GradedRingForHomalg Package	5
3	Quick Start	6
4	Graded Rings	7
4.1	Graded Rings: Category and Representations	7
4.2	Graded Rings: Constructors	8
4.3	Graded Rings: Attributes	8
4.4	Graded Rings: Operations and Functions	9
5	Homogeneous Matrices	10
5.1	Homogeneous Matrices: Category and Representations	10
5.2	Homogeneous Matrices: Constructors	10
5.3	Homogeneous Matrices: Attributes	11
5.4	Homogeneous Matrices: Operations and Functions	11
6	Examples	13
A	The Matrix Tool Operations	14
A.1	The Tool Operations <i>without</i> a Fallback Method	14
A.2	The Tool Operations with a Fallback Method	14
B	Overview of the GradedRingForHomalg Package Source Code	17
B.1	The generic Methods	17
B.2	Tools	18
	References	20
	Index	21

Chapter 1

Introduction

1.1 What is the Role of the `GradedRingForHomalg` Package in the `homalg` Project?

The `homalg` project [[hpa10](#)] aims at providing a general and abstract framework for homological computations. The package `GradedRingForHomalg` enables the `homalg` project to endow commutative rings in `homalg` with an Abelian grading.

1.2 Functionality

The package `GradedRingForHomalg` on the one hand builds on the package `MatricesForHomalg` and on the other hands adds functionality to `MatricesForHomalg`.

1.3 The Math Behind This Package

Chapter 2

Installation of the GradedRingForHomalg Package

To install this package just extract the package's archive file to the GAP pkg directory. GradedRingForHomalg also needs the package homalg.

By default the GradedRingForHomalg package is not automatically loaded by GAP when it is installed. You must load the package with

```
LoadPackage("GradedRingForHomalg");
```

before its functions become available.

Please, send me us e-mail if you have any questions, remarks, suggestions, etc. concerning this package. Also, we would be pleased to hear about applications of this package.

Mohamed Barakat and Markus Lange-Hegermann

Chapter 3

Quick Start

Chapter 4

Graded Rings

The package `GradedRingForHomalg` defines the classes of graded rings, ring elements and matrices over such rings. These three objects can be used as data structures defined in `MatricesForHomalg` on which the `homalg` project can rely to do homological computations over graded rings.

The graded rings most prominently can be used with methods known from general `homalg` rings. The methods for doing the computations are presented in the appendix (B), since they are not for external use. The new attributes and operations are documented here.

Since the objects implemented here are representations from objects elsewhere in the `homalg` project (i.e. `MatricesForHomalg`), we want to stress that there are many other operations in `MatricesForHomalg`, which can be used in connection with the ones presented here. A few of them can be found in the examples and the appendix of this documentation.

4.1 Graded Rings: Category and Representations

4.1.1 `IsHomalgGradedRingRep`

▷ `IsHomalgGradedRingRep(R)` (Representation)

Returns: true or false

The representation of `homalg` graded rings.

(It is a subrepresentation of the `GAP` representation

`IsHomalgRingOrFinitelyPresentedModuleRep`.)

```
Code
DeclareRepresentation( "IsHomalgGradedRingRep",
  IsHomalgGradedRing and
  IsHomalgGradedRingOrGradedModuleRep,
  [ "ring" ] );
```

4.1.2 `IsHomalgGradedRingElementRep`

▷ `IsHomalgGradedRingElementRep(r)` (Representation)

Returns: true or false

The representation of elements of `homalg` graded rings.

(It is a representation of the `GAP` category `IsHomalgRingElement`.)


```

Code
DeclareRepresentation( "IsHomalgGradedRingElementRep",
    IsHomalgGradedRingElement,
    [ ] );

```

4.2 Graded Rings: Constructors

4.2.1 HomalgGradedRingElement (constructor for graded ring elements using numerator and denominator)

- ▷ HomalgGradedRingElement(*numer*, *denom*, *R*) (function)
- ▷ HomalgGradedRingElement(*numer*, *R*) (function)

Returns: a graded ring element

Creates the graded ring element $\text{numer}/\text{denom}$ or in the second case $\text{numer}/1$ for the graded ring R . Both *numer* and *denom* may either be a string describing a valid global ring element or from the global ring or computation ring.

4.3 Graded Rings: Attributes

4.3.1 DegreeGroup

- ▷ DegreeGroup(*S*) (attribute)

Returns: a left \mathbb{Z} -module

The degree Abelian group of the commutative graded ring S .

4.3.2 CommonNonTrivialWeightOfIndeterminates

- ▷ CommonNonTrivialWeightOfIndeterminates(*S*) (attribute)

Returns: a degree

The common nontrivial weight of the indeterminates of the graded ring S if it exists. Otherwise an error is issued. WARNING: Since the DegreeGroup and WeightsOfIndeterminates are in some cases bound together, you MUST not set the DegreeGroup by hand and let the algorithm create the weights. Set both by hand, set only weights or use the method WeightsOfIndeterminates to set both. Never set the DegreeGroup without the WeightsOfIndeterminates, because it simply wont work!

4.3.3 WeightsOfIndeterminates

- ▷ WeightsOfIndeterminates(*S*) (attribute)

Returns: a list or listlist of integers

The list of degrees of the indeterminates of the graded ring S .

4.3.4 MatrixOfWeightsOfIndeterminates

- ▷ MatrixOfWeightsOfIndeterminates(*S*) (attribute)

Returns: a homalg matrix

A homalg matrix where the list (or listlist) of degrees of the indeterminates of the graded ring S is stored.

4.4 Graded Rings: Operations and Functions

4.4.1 UnderlyingNonGradedRing (for homalg graded rings)

- ▷ `UnderlyingNonGradedRing(R)` (operation)
Returns: a homalg ring
Internally there is a ring, in which computations take place.

4.4.2 UnderlyingNonGradedRing (for homalg graded ring elements)

- ▷ `UnderlyingNonGradedRing(r)` (operation)
Returns: a homalg ring
Internally there is a ring, in which computations take place.

4.4.3 Name (for homalg graded ring elements)

- ▷ `Name(r)` (operation)
Returns: a string
The name of the graded ring element r .

Chapter 5

Homogeneous Matrices

The package `GradedRingForHomalg` defines the classes of graded rings, ring elements and homogeneous matrices over such rings. These three objects can be used as data structures defined in `MatricesForHomalg` on which the `homalg` project can rely to do homological computations over graded rings.

The graded rings most prominently can be used with methods known from general `homalg` rings. The methods for doing the computations are presented in the appendix (B), since they are not for external use. The new attributes and operations are documented here.

Since the objects implemented here are representations from objects elsewhere in the `homalg` project (i.e. `MatricesForHomalg`), we want to stress that there are many other operations in `MatricesForHomalg`, which can be used in connection with the ones presented here. A few of them can be found in the examples and the appendix of this documentation.

5.1 Homogeneous Matrices: Category and Representations

5.1.1 IsHomalgMatrixOverGradedRingRep

▷ `IsHomalgMatrixOverGradedRingRep(A)` (Representation)

Returns: true or false

The representation of `homalg` matrices with entries in a `homalg` graded ring.

(It is a representation of the GAP category `IsMatrixOverGradedRing`.)

Code

```
DeclareRepresentation( "IsHomalgMatrixOverGradedRingRep",  
  IsMatrixOverGradedRing,  
  [ ] );
```

5.2 Homogeneous Matrices: Constructors

5.2.1 MatrixOverGradedRing (constructor for matrices over graded rings)

▷ `MatrixOverGradedRing(mat, S)` (function)

Returns: a matrix over a graded ring

Creates a matrix for the graded ring S , where mat is a matrix over `UnderlyingNonGradedRing(S)`.

5.3 Homogeneous Matrices: Attributes

5.3.1 DegreesOfEntries

- ▷ `DegreesOfEntries(A)` (attribute)
Returns: a listlist of degrees/multi-degrees
 The matrix of degrees of the matrix A .

5.3.2 NonTrivialDegreePerRow

- ▷ `NonTrivialDegreePerRow(A[, col_degrees])` (attribute)
Returns: a list of degrees/multi-degrees
 The list of first nontrivial degree per row of the matrix A .

5.3.3 NonTrivialDegreePerColumn

- ▷ `NonTrivialDegreePerColumn(A[, row_degrees])` (attribute)
Returns: a list of degrees/multi-degrees
 The list of first nontrivial degree per column of the matrix A .

5.4 Homogeneous Matrices: Operations and Functions

5.4.1 UnderlyingNonGradedRing (for matrices over graded rings)

- ▷ `UnderlyingNonGradedRing(mat)` (operation)
Returns: a homalg ring
 The nongraded ring underlying `HomalgRing(mat)`.

5.4.2 SetMatElm (for matrices over graded rings)

- ▷ `SetMatElm(mat, i, j, r, R)` (operation)

Changes the entry (i, j) of the matrix mat to the value r . Here R is the graded homalg ring involved in these computations.

5.4.3 AddToMatElm (for matrices over graded rings)

- ▷ `AddToMatElm(mat, i, j, r, R)` (operation)

Changes the entry (i, j) of the matrix mat by adding the value r to it. Here R is the (graded) homalg ring involved in these computations.

5.4.4 MatElmAsString (for matrices over graded rings)

- ▷ `MatElmAsString(mat, i, j, R)` (operation)
Returns: a string
 Returns the entry (i, j) of the matrix mat as a string. Here R is the (graded) homalg ring involved in these computations.

5.4.5 MatElm (for matrices over graded rings)

▷ `MatElm(mat, i, j, R)`

(operation)

Returns: a graded ring element

Returns the entry (i, j) of the matrix *mat*. Here *R* is the (graded) homalg ring involved in these computations.

Chapter 6

Examples

Appendix A

The Matrix Tool Operations

The functions listed below are components of the `homalgTable` object stored in the ring. They are only indirectly accessible through standard methods that invoke them.

A.1 The Tool Operations *without* a Fallback Method

There are matrix methods for which `homalg` needs a `homalgTable` entry for non-internal rings, as it cannot provide a suitable fallback. Below is the list of these `homalgTable` entries.

A.2 The Tool Operations with a Fallback Method

These are the methods for which it is recommended for performance reasons to have a `homalgTable` entry for non-internal rings. `homalg` only provides a generic fallback method.

A.2.1 MonomialMatrix

▷ `MonomialMatrix(d , R)` (operation)

Returns: a `homalg` matrix

The column matrix of d -th monomials of the `homalg` graded ring R .

Example

```
gap> R := HomalgFieldOfRationalsInDefaultCAS( ) * "x,y,z";
gap> S := GradedRing( R );
gap> m := MonomialMatrix( 2, S );
<A ? x 1 matrix over a graded ring>
gap> NrRows( m );
6
gap> m;
<A 6 x 1 matrix over a graded ring>
gap> Display( m );
x^2,
x*y,
x*z,
y^2,
y*z,
z^2
(over a graded ring)
```

A.2.2 RandomMatrixBetweenGradedFreeLeftModules

▷ RandomMatrixBetweenGradedFreeLeftModules(*degreesS*, *degreesT*, *R*) (operation)

Returns: a homalg matrix

A random $r \times c$ -matrix between the graded free *left* modules $R^{(-\text{degreesS})} \rightarrow R^{(-\text{degreesT})}$, where $r = \text{Length}(\text{degreesS})$ and $c = \text{Length}(\text{degreesT})$.

Example

```
gap> R := HomalgFieldOfRationalsInDefaultCAS( ) * "a,b,c";;
gap> S := GradedRing( R );;
gap> rand := RandomMatrixBetweenGradedFreeLeftModules( [ 2, 3, 4 ], [ 1, 2 ], S );
<A 3 x 2 matrix over a graded ring>
gap> #Display( rand );
gap> #a-2*b+2*c, 2,
gap> #a^2-a*b+b^2-2*b*c+5*c^2, 3*c,
gap> #2*a^3-3*a^2*b+2*a*b^2+3*a^2*c+a*b*c-2*b^2*c-3*b*c^2-2*c^3,a^2-4*a*b-3*a*c-c^2
```

A.2.3 RandomMatrixBetweenGradedFreeRightModules

▷ RandomMatrixBetweenGradedFreeRightModules(*degreesT*, *degreesS*, *R*) (operation)

Returns: a homalg matrix

A random $r \times c$ -matrix between the graded free *right* modules $R^{(-\text{degreesS})} \rightarrow R^{(-\text{degreesT})}$, where $r = \text{Length}(\text{degreesT})$ and $c = \text{Length}(\text{degreesS})$.

Example

```
gap> R := HomalgFieldOfRationalsInDefaultCAS( ) * "a,b,c";;
gap> S := GradedRing( R );;
gap> rand := RandomMatrixBetweenGradedFreeRightModules( [ 1, 2 ], [ 2, 3, 4 ], S );
<A 2 x 3 matrix over a graded ring>
gap> #Display( rand );
gap> #a-2*b-c,a*b+b^2-b*c,2*a^3-a*b^2-4*b^3+4*a^2*c-3*a*b*c-b^2*c+a*c^2+5*b*c^2-2*c^3,
gap> #-5, -2*a+c, -2*a^2-a*b-2*b^2-3*a*c
```

A.2.4 Diff

▷ Diff(*D*, *N*) (operation)

Returns: a homalg matrix

If D is a $f \times p$ -matrix and N is a $g \times q$ -matrix then $H = \text{Diff}(D, N)$ is an $fg \times pq$ -matrix whose entry $H[g * (i - 1) + j, q * (k - 1) + l]$ is the result of differentiating $N[j, l]$ by the differential operator corresponding to $D[i, k]$. (Here we follow the Macaulay2 convention.)

Example

```
gap> S := HomalgFieldOfRationalsInDefaultCAS( ) * "a,b,c" * "x,y,z";;
gap> D := HomalgMatrix( "[ \
> x,2*y, \
> y,a-b^2, \
> z,y-b \
> ]", 3, 2, S );
<A 3 x 2 matrix over an external ring>
gap> N := HomalgMatrix( "[ \
> x^2-a*y^3,x^3-z^2*y,x*y-b,x*z-c, \
> x, x*y, a-b, x*a*b \
> ]", 2, 4, S );
```



```

<A 2 x 4 matrix over an external ring>
gap> H := Diff( D, N );
<A 6 x 8 matrix over an external ring>
gap> Display( H );
2*x,      3*x^2, y,z,  -6*a*y^2,-2*z^2,2*x,0,
1,        y,      0,a*b,0,      2*x,  0,  0,
-3*a*y^2,-z^2,  x,0,  -y^3,   0,    0,  0,
0,        x,      0,0,  0,      0,    1,  b*x,
0,        -2*y*z,0,x,  -3*a*y^2,-z^2,  x+1,0,
0,        0,      0,0,  0,      x,    1,  -a*x

```

Appendix B

Overview of the GradedRingForHomalg Package Source Code

This appendix is included in the documentation to shine some light on the mathematical backgrounds of this Package. Neither is it needed to work with this package nor should the methods presented here be called directly. The functions documented here are entries of the so called ring table and not to be called directly. There are higher level methods in declared and installed in `MatricesForHomalg`, which call this functions (\rightarrow ?MatricesForHomalg:The Basic Matrix Operations).

B.1 The generic Methods

We will present some methods as an example, to show the idea:

B.1.1 BasisOfRowModule (for graded rings)

▷ `BasisOfRowModule(M)` (function)
Returns: a distinguished basis (i.e. a distinguished generating set) of the module generated by `M`

```
Code
BasisOfRowModule :=
  function( M )
    return MatrixOverGradedRing(
      BasisOfRowModule( UnderlyingMatrixOverNonGradedRing( M ) ),
      HomalgRing( M ) );
  end,
```

B.1.2 DecideZeroRows (for graded rings)

▷ `DecideZeroRows(A, B)` (function)
Returns: a reduced form of `A` with respect to `B`

```
Code
DecideZeroRows :=
  function( A, B )
```

```

return MatrixOverGradedRing(
    DecideZeroRows( UnderlyingMatrixOverNonGradedRing( A ),
                    UnderlyingMatrixOverNonGradedRing( B ) ),
    HomalgRing( A ) );
end,

```

B.1.3 SyzygiesGeneratorsOfRows (for graded rings)

▷ SyzygiesGeneratorsOfRows(M)

(function)

Returns: a distinguished basis of the syzygies of the argument

Code

```

SyzygiesGeneratorsOfRows :=
function( M )
    return MatrixOverGradedRing(
        SyzygiesGeneratorsOfRows( UnderlyingMatrixOverNonGradedRing( M ) ),
        HomalgRing( M ) );
end,

```

B.2 Tools

The package GradedRingForHomalg also implements tool functions. These are referred to from MatricesForHomalg automatically. We list the implemented methods here and refer to the MatricesForHomalg documentation (\rightarrow ?MatricesForHomalg: The Matrix Tool Operations and ?MatricesForHomalg:RingElement) for details. All tools functions from MatricesForHomalg not listed here are also supported by fallback tools.

- IsZero
- IsOne
- Minus
- DivideByUnit
- IsUnit
- Sum
- Product
- ShallowCopy
- ZeroMatrix
- IdentityMatrix
- AreEqualMatrices
- Involution
- CertainRows

- CertainColumns
- UnionOfRows
- UnionOfColumns
- DiagMat
- KroneckerMat
- MulMat
- AddMat
- SubMat
- Compose
- NrRows
- NrColumns
- IsZeroMatrix
- IsDiagonalMatrix
- ZeroRows
- ZeroColumns

References

- [hpa10] T. homalg project authors. *The homalg project*, 2003-2010. <http://homalg.math.rwth-aachen.de/>. 4

Index

- GradedRingForHomalg, [4](#)
- AddToMatElm
 - for matrices over graded rings, [11](#)
- BasisOfRowModule
 - for graded rings, [17](#)
- CommonNonTrivialWeightOfIndeterminates, [8](#)
- DecideZeroRows
 - for graded rings, [17](#)
- DegreeGroup, [8](#)
- DegreesOfEntries, [11](#)
- Diff, [15](#)
- HomalgGradedRingElement
 - constructor for graded ring elements using a given numerator and one as denominator, [8](#)
 - constructor for graded ring elements using numerator and denominator, [8](#)
- IsHomalgGradedRingElementRep, [7](#)
- IsHomalgGradedRingRep, [7](#)
- IsHomalgMatrixOverGradedRingRep, [10](#)
- MatElm
 - for matrices over graded rings, [12](#)
- MatElmAsString
 - for matrices over graded rings, [11](#)
- MatrixOfWeightsOfIndeterminates, [8](#)
- MatrixOverGradedRing
 - constructor for matrices over graded rings, [10](#)
- MonomialMatrix, [14](#)
- Name
 - for homalg graded ring elements, [9](#)
- NonTrivialDegreePerColumn, [11](#)
- NonTrivialDegreePerRow, [11](#)
- RandomMatrixBetweenGradedFreeLeftModules, [15](#)
- RandomMatrixBetweenGradedFreeRightModules, [15](#)
- SetMatElm
 - for matrices over graded rings, [11](#)
- SyzygiesGeneratorsOfRows
 - for graded rings, [18](#)
- UnderlyingNonGradedRing
 - for homalg graded ring elements, [9](#)
 - for homalg graded rings, [9](#)
 - for matrices over graded rings, [11](#)
- WeightsOfIndeterminates, [8](#)