ToolsForFPGradedModules

A package to provide additional structures for toric varieties

2020.01.16

16 January 2020

Martin Bies

Martin Bies

Email: martin.bies@alumni.uni-heidelberg.de

Homepage: https://martinbies.github.io/

Address: Mathematical Institute
University of Oxford
Andrew Wiles Building
Radcliffe Observatory Quarter
Woodstock Road
Oxford OX2 6GG
United Kingdom

Contents

Chapter 1

Introduction

1.1 What is the goal of the ToolsForFPGradedModules package?

ToolsForFPGradedModules provides additional tools to perform computations or manipulate FPGradedModules, which are for structural reasons not part of the underlying package for *FreydCategories*.

Chapter 2

Tools for FPGradedModules

2.1 Minimal free resolutions

2.1.1 LeftIdealForCAP (for IsList, IsHomalgGradedRing)

▷ `LeftIdealForCAP(L, R)` (operation)

Returns: a f.p. module presentation

The argument is a list L of generators of an ideal and a homalg graded ring R. This method then construct the left ideal in this ring generated by these generators.

2.1.2 RightIdealForCAP (for IsList, IsHomalgGradedRing)

▷ `RightIdealForCAP(L, R)` (operation)

Returns: a f.p. module presentation

The argument is a list L of generators of an ideal and a homalg graded ring R. This method then construct the right ideal in this ring generated by these generators.

2.1.3 MinimalFreeResolutionForCAP (for IsFpGradedLeftOrRightModulesObject)

▷ `MinimalFreeResolutionForCAP(M)` (attribute)

Returns: a complex of projective graded module morphisms

The argument is a graded left or right module presentation *M*. We then compute a minimal free resolution of *M*.

2.2 Betti tables

2.2.1 BettiTableForCAP (for IsFpGradedLeftOrRightModulesObject)

▷ `BettiTableForCAP(M)` (attribute)

Returns: a list of lists

The argument is a graded left or right module presentation *M*. We then compute the Betti table of *M*.

2.3 Example: Minimal free resolution and Betti table

```
gap> Q := HomalgFieldOfRationalsInSingular();
Q
gap> S := GradedRing( Q * "x_1, x_2, x_3" );
Q[x_1,x_2,x_3]
(weights: yet unset)
gap> SetWeightsOfIndeterminates( S, [[1],[1],[1]] );

gap> vars := IndeterminatesOfPolynomialRing( S );;
gap> IR := LeftIdealForCAP( [ vars[ 1 ], vars[ 2 ], vars[ 3 ] ], S );;
gap> IsWellDefined( IR );
true
gap> resolution := MinimalFreeResolutionForCAP( IR );
<An object in Complex category of Category of graded
rows over Q[x_1,x_2,x_3] (with weights [ 1, 1, 1 ])>
gap> differential_function :=
>                      UnderlyingZFunctorCell( resolution )!.differential_func;
function( i ) ... end
gap> IsWellDefined( differential_function( -1 ) );
true
gap> IsWellDefined( differential_function( -2 ) );
true
gap> IsWellDefined( differential_function( -3 ) );
true
gap> BT := BettiTableForCAP( IR );
[ [ -1, -1, -1 ], [ -2, -2, -2 ], [ -3 ] ]
```

Chapter 3

Truncations of GradedExt for f.p. graded modules

3.1 Truncations of InternalHoms of FpGradedModules

3.1.1 TruncateInternalHom (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHom(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.1.2 TruncateInternalHomEmbedding (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomEmbedding(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.1.3 TruncateInternalHom (for IsToricVariety, IsFpGradedLeftOrRightModulesMorphism, IsFpGradedLeftOrRightModulesMorphism, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHom(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.2 Truncations of InternalHoms of FpGradedModules to degree zero

3.2.1 TruncateInternalHomToZero (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomToZero(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.2.2 TruncateInternalHomEmbeddingToZero (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomEmbeddingToZero(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.2.3 TruncateInternalHomToZero (for IsToricVariety, IsFpGradedLeftOrRightModulesMorphism, IsFpGradedLeftOrRightModulesMorphism, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomToZero(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.3 Truncations of InternalHoms of FpGradedModules in parallel

3.3.1 TruncateInternalHomInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomInParallel(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.3.2 TruncateInternalHomEmbeddingInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomEmbeddingInParallel(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.3.3 TruncateInternalHomInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesMorphism, IsFpGradedLeftOrRightModulesMorphism, IsList, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomInParallel(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.4 Truncations of InternalHoms of FpGradedModules to degree zero in parallel

3.4.1 TruncateInternalHomToZeroInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomToZeroInParallel(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.4.2 TruncateInternalHomEmbeddingToZeroInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomEmbeddingToZeroInParallel(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.4.3 TruncateInternalHomToZeroInParallel (for IsToricVariety, IsFpGradedLeftOrRightModulesMorphism, IsFpGradedLeftOrRightModulesMorphism, IsBool, IsFieldForHomalg)

▷ `TruncateInternalHomToZeroInParallel(`*arg1, arg2, arg3, arg4, arg5*`)` (operation)

3.4.4 TruncateGradedExt (for IsInt, IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsList)

▷ `TruncateGradedExt(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.4.5 TruncateGradedExtToZero (for IsInt, IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateGradedExtToZero(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.4.6 TruncateGradedExtInParallel (for IsInt, IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsList, IsList)

▷ `TruncateGradedExtInParallel(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.4.7 TruncateGradedExtToZeroInParallel (for IsInt, IsToricVariety, IsFpGradedLeftOrRightModulesObject, IsFpGradedLeftOrRightModulesObject, IsBool, IsFieldForHomalg)

▷ `TruncateGradedExtToZeroInParallel(`*arg1, arg2, arg3, arg4, arg5, arg6*`)` (operation)

3.5 Examples

3.5.1 Truncation of IntHom

Example

```
gap> P2 := ProjectiveSpace( 2 );
<A projective toric variety of dimension 2>
gap> cox_ring := CoxRing( P2 );
```

```
Q[x_1,x_2,x_3]
(weights: [ 1, 1, 1 ])
gap> source := GradedRow( [[[-1],1]], cox_ring );
<A graded row of rank 1>
gap> range := GradedRow( [[[0],1]], cox_ring );
<A graded row of rank 1>
gap> vars := IndeterminatesOfPolynomialRing( cox_ring );;
gap> matrix := HomalgMatrix( [[ vars[ 1 ] ]], cox_ring );
<A 1 x 1 matrix over a graded ring>
gap> obj1 := FreydCategoryObject(
>           GradedRowOrColumnMorphism( source, matrix, range ) );
<An object in Category of f.p. graded
left modules over Q[x_1,x_2,x_3]
(with weights [ 1, 1, 1 ])>
gap> IsWellDefined( obj1 );
true
gap> source := GradedRow( [[[-1],1]], cox_ring );
<A graded row of rank 1>
gap> range := GradedRow( [[[1],2]], cox_ring );
<A graded row of rank 2>
gap> matrix := HomalgMatrix( [[ vars[ 1 ] * vars[ 2 ],
>                               vars[ 1 ] * vars[ 3 ] ]], cox_ring );
<A 1 x 2 matrix over a graded ring>
gap> obj2 := FreydCategoryObject(
>           GradedRowOrColumnMorphism( source, matrix, range ) );
<An object in Category of f.p. graded
left modules over Q[x_1,x_2,x_3]
(with weights [ 1, 1, 1 ])>
gap> IsWellDefined( obj2 );
true
gap> source := GradedRow( [[[0],1]], cox_ring );
<A graded row of rank 1>
gap> range := GradedRow( [[[1],2]], cox_ring );
<A graded row of rank 2>
gap> matrix := HomalgMatrix( [[ vars[ 2 ], vars[ 3 ] ]], cox_ring );
<A 1 x 2 matrix over a graded ring>
gap> mor := GradedRowOrColumnMorphism( source, matrix, range );
<A morphism in Category of graded rows
over Q[x_1,x_2,x_3] (with weights [ 1, 1, 1 ])>
gap> pres_mor := FreydCategoryMorphism( obj1, mor, obj2 );
<A morphism in Category of f.p. graded
left modules over Q[x_1,x_2,x_3]
(with weights [ 1, 1, 1 ])>
gap> IsWellDefined( pres_mor );
true
gap> Q := HomalgFieldOfRationalsInSingular();
Q
gap> m1 := TruncateInternalHom( P2, obj1, obj2, [ 4 ], false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m1 );
true
gap> m2 := TruncateInternalHomEmbedding( P2, obj1, obj2, [ 4 ], false, Q );
```

```
<A monomorphism in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m2 );
true
gap> m3 := TruncateInternalHom( P2, pres_mor, IdentityMorphism( obj2 ), [ 4 ], false, Q );
<A morphism in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m3 );
true
```

3.5.2 Truncation of IntHom to degree zero

Example

```
gap> m4 := TruncateInternalHomToZero( P2, obj1, obj2, false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m4 );
true
gap> m5 := TruncateInternalHomEmbeddingToZero( P2, obj1, obj2, false, Q );
<A monomorphism in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m5 );
true
gap> m6 := TruncateInternalHomToZero( P2, pres_mor, IdentityMorphism( obj2 ), false, Q );
<A morphism in Freyd( Category of matrices over Q )>
gap> IsWellDefined( m6 );
true
```

3.5.3 Truncation of IntHom in parallel

Example

```
gap> m7 := TruncateInternalHomInParallel( P2, obj1, obj2, [ 4 ], false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> m1 = m7;
true
gap> m8 := TruncateInternalHomEmbeddingInParallel( P2, obj1, obj2, [ 4 ], false, Q );
<A monomorphism in Freyd( Category of matrices over Q )>
gap> m8 = m2;
true
gap> m9 := TruncateInternalHomInParallel( P2, pres_mor, IdentityMorphism( obj2 ), [ 4 ], false, Q
<A morphism in Freyd( Category of matrices over Q )>
gap> m9 = m3;
true
```

3.5.4 Truncation of IntHom to degree zero in parallel

Example

```
gap> m10 := TruncateInternalHomToZeroInParallel( P2, obj1, obj2, false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> m10 = m4;
true
gap> m11 := TruncateInternalHomEmbeddingToZeroInParallel( P2, obj1, obj2, false, Q );
<A monomorphism in Freyd( Category of matrices over Q )>
gap> m11 = m5;
true
gap> m12 := TruncateInternalHomToZeroInParallel( P2, pres_mor, IdentityMorphism( obj2 ), false, Q
<A morphism in Freyd( Category of matrices over Q )>
```

```
gap> m12 = m6;
true
```

3.5.5 Truncation of GradedExt

Example

```
gap> v1 := TruncateGradedExt( 1, P2, obj1, obj2, [ 4 ], [ false, Q ] );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( v1 );
true
gap> v2 := TruncateGradedExt( 1, P2, obj1, obj2, [ 0 ], [ false, Q ] );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( v2 );
true
gap> v3 := TruncateGradedExtToZero( 1, P2, obj1, obj2, false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> v3 = v2;
true
gap> v4 := TruncateGradedExtInParallel( 1, P2, obj1, obj2, [ 4 ], [ false, Q ] );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( v4 );
true
gap> v5 := TruncateGradedExtInParallel( 1, P2, obj1, obj2, [ 0 ], [ false, Q ] );
<An object in Freyd( Category of matrices over Q )>
gap> IsWellDefined( v5 );
true
gap> v6 := TruncateGradedExtToZeroInParallel( 1, P2, obj1, obj2, false, Q );
<An object in Freyd( Category of matrices over Q )>
gap> v6 = v5;
true
```

Index