### **Bicomplexes for Abelian categories**

2017.05.02

2 May 2017

#### **Mohamed Barakat**

#### **Kamal Saleh**

#### **Mohamed Barakat**

Email: mohamed.barakat@uni-siegen.de

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

Address: Walter-Flex-Str. 3 57068 Siegen Germany

#### **Kamal Saleh**

Email: kamal.saleh@uni-siegen.de

Homepage: https://github.com/kamalsaleh/

Address: Walter-Flex-Str. 3 57068 Siegen Germany

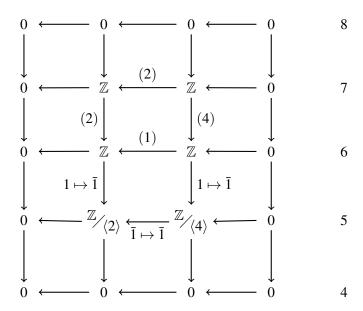
### **Contents**

1	Bicomplexes			
	1.1	Categories	7	
		Constructors		
	1.3	Attributes	ç	
In	dex		1(	

### Chapter 1

### **Bicomplexes**

Let us create the following chain complex of chain complexes of left presentations over  $\mathbb{Z}$ :

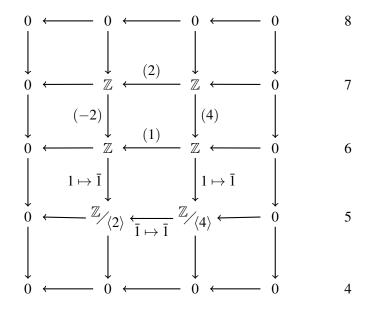


$$\mathbf{0} \longleftarrow C_9 \stackrel{\phi}{\longleftarrow} C_{10} \longleftarrow \mathbf{0}$$

```
gap> ZZ := HomalgRingOfIntegers();
Z
gap> lp_cat := CategoryOfHomalgLeftModules( ZZ );
intrinsic Category of left presentations of Z with ambient objects
gap> chains_lp_cat := ChainComplexCategory( lp_cat );
Chain complexes category over intrinsic Category
of left presentations of Z with ambient objects
gap> chains_chains_lp_cat := ChainComplexCategory( chains_lp_cat );
Chain complexes category over chain complexes category over
intrinsic Category of left presentations of Z with ambient objects
gap> Bicomplexes_cat := AsCategoryOfBicomplexes( chains_chains_lp_cat );
Chain complexes category over chain complexes category over
```

```
intrinsic Category of left presentations of Z with ambient objects
as bicomplexes
gap> F1 := HomalgFreeLeftModule( 1, ZZ );
<A free left module of rank 1 on a free generator>
gap> d7 := HomalgMap( HomalgMatrix( "[ [ 4 ] ]", 1, 1, ZZ ), F1, F1 );
<An endo"morphism" of a left module>
gap> d6 := CokernelProjection( d7 );
<An epimorphism of left modules>
gap> C10 := ChainComplex( [ d6, d7 ], 6 );
<A bounded object in chain complexes category over intrinsic</pre>
Category of left presentations of Z with ambient objects with
active lower bound 4 and active upper bound 8>
gap> t7 := HomalgMap( HomalgMatrix( "[ [ 2 ] ]", 1, 1, ZZ ), F1, F1 );
<An endo"morphism" of a left module>
gap> t6 := CokernelProjection( t7 );
<An epimorphism of left modules>
gap> C9 := ChainComplex( [ t6, t7 ], 6 );
<A bounded object in chain complexes category over intrinsic</pre>
Category of left presentations of Z with ambient objects with
active lower bound 4 and active upper bound 8>
gap> phi5 := HomalgMap( HomalgIdentityMatrix( 1, ZZ ), C10[ 5 ], C9[ 5 ] );
<A "homomorphism" of left modules>
gap> phi6 := HomalgMap( HomalgIdentityMatrix( 1, ZZ ), F1, F1 );
<An endo"morphism" of a left module>
gap> phi7 := HomalgMap( 2 * HomalgIdentityMatrix( 1, ZZ ), F1, F1 );
<An endo"morphism" of a left module>
gap> phi := ChainMorphism( C10, C9, [ phi5, phi6, phi7 ], 5 );
<A bounded morphism in chain complexes category over intrinsic</pre>
Category of left presentations of Z with ambient objects with
active lower bound 4 and active upper bound 8>
gap> C := ChainComplex( [ phi ], 10 );
<A bounded object in chain complexes category over chain complexes</pre>
category over intrinsic Category of left presentations of Z with
ambient objects with active lower bound 8 and active upper bound 11>
```

Now we compute its associated homological bicomplex B and the total complex of B:



8 9 10 11

```
_ Example _
gap> B := HomologicalBicomplex( C );
<A homological bicomplex in intrinsic Category of left presentations
of Z with ambient objects concentrated in window [ 8 .. 11 ] x [ 4 .. 8 ]>
gap> Display( VerticalDifferentialAt( B, 9, 7 ) );
[ [ -2 ] ]
the map is currently represented by the above 1 x 1 matrix
gap> T := TotalComplex( B );
<A bounded object in chain complexes category over intrinsic Category</pre>
of left presentations of Z with ambient objects with active lower
bound 13 and active upper bound 18>
gap> T[ 13 ];
<A zero left module>
gap> T[ 14 ];
<A cyclic torsion left module presented by 1 relation</pre>
for a cyclic generator>
gap> T[ 15 ];
<A non-torsion left module presented by 1 relation for 2 generators>
gap> T[ 16 ];
<A free left module of rank 2 on free generators>
gap> T[ 17 ];
<A free left module of rank 1 on a free generator>
gap> T[ 18 ];
<A zero left module>
gap> Display( T^16 );
[[-2, 0],
           1]
     1,
the map is currently represented by the above 2 x 2 matrix
```

```
gap> IsExact( T );
true
gap> T;
Category of left presentations of Z with ambient objects with active
lower bound 13 and active upper bound 18>
gap> Display( T, 13, 18 );
______
In index 13
Object is
Differential is
(an empty 0 x 0 matrix)
the map is currently represented by the above 0 x 0 matrix
______
In index 14
Object is
Z/< 2 >
Differential is
(an empty 1 x 0 matrix)
the map is currently represented by the above 1 x 0 matrix
_____
In index 15
Object is
[[0,4]]
Cokernel of the map
Z^{(1x1)} \longrightarrow Z^{(1x2)}
currently represented by the above matrix
Differential is
[[-1],
 [ 1]]
the map is currently represented by the above 2 \times 1 matrix
In index 16
Object is
Z^{(1 \times 2)}
```

```
Differential is
[[ -2, 0],
         1]]
    1,
the map is currently represented by the above 2 x 2 matrix
In index 17
Object is
Z^{(1 \times 1)}
Differential is
[[2, 4]]
the map is currently represented by the above 1 x 2 matrix
In index 18
Object is
Differential is
(an empty 0 x 1 matrix)
the map is currently represented by the above 0 \times 1 matrix
```

#### 1.1 Categories

### 1.1.1 IsCapCategoryBicomplexCell (for IsCapCategoryCell andIsAttributeStoringRep)

▷ IsCapCategoryBicomplexCell(arg)

(filter)

Returns: true or false

The GAP category of cells in a CAP category of bicomplexes.

# 1.1.2 IsCapCategoryBicomplexObject (for IsCapCategoryBicomplexCell andIsCapCategoryObject)

▷ IsCapCategoryBicomplexObject(arg)

(filter)

Returns: true or false

The GAP category of bicomplex objects in a CAP category of bicomplexes.

# 1.1.3 IsCapCategoryHomologicalBicomplexObject (for IsCapCategoryBicomplexObject)

▷ IsCapCategoryHomologicalBicomplexObject(arg)

(filter)

Returns: true or false

The GAP category of homological bicomplex objects in a CAP category of bicomplexes.

### 1.1.4 IsCapCategoryCohomologicalBicomplexObject (for IsCapCategoryBicomplexObject)

▷ IsCapCategoryCohomologicalBicomplexObject(arg)

(filter)

Returns: true or false

The GAP category of cohomological bicomplex objects in a CAP category of bicomplexes.

#### 1.1.5 IsCapCategoryBicomplexMorphism (for IsCapCategoryBicomplexCell andIs-CapCategoryMorphism)

▷ IsCapCategoryBicomplexMorphism(arg)

(filter)

Returns: true or false

The GAP category of bicomplex morphisms in a CAP category of bicomplexes.

# ${\bf 1.1.6} \quad Is Cap Category Homological Bicomplex Morphism \quad (for \quad Is Cap Category Bicomplex Morphism)$

▷ IsCapCategoryHomologicalBicomplexMorphism(arg)

(filter)

Returns: true or false

The GAP category of homological bicomplex morphisms in a CAP category of bicomplexes.

### 1.1.7 IsCapCategoryCohomologicalBicomplexMorphism (for IsCapCategoryBicomplexMorphism)

▷ IsCapCategoryCohomologicalBicomplexMorphism(arg)

(filter)

Returns: true or false

The GAP category of cohomological bicomplex morphisms in a CAP category of bicomplexes.

#### 1.2 Constructors

#### 1.2.1 AssociatedBicomplexObject (for IsChainOrCochainComplex)

▷ AssociatedBicomplexObject(cx)

(attribute)

**Returns:** a CAP object

Return the bicomplex associated to the complex of complexes cx.

#### 1.2.2 AssociatedBicomplexMorphism (for IsChainOrCochainMorphism)

▷ AssociatedBicomplexMorphism(mu)

(attribute)

**Returns:** a CAP morphism

Return the morphism of bicomplexes associated to the chain morphism between two complexes of complexes mu.

#### 1.2.3 AssociatedBicomplexFunctor (for IsCapFunctor, IsString)

▷ AssociatedBicomplexFunctor(F, name)

(operation)

▷ AssociatedBicomplexFunctor(F)

(attribute)

**Returns:** a CAP functor

#### 1.2.4 AssociatedBicomplex (for IsCapNaturalTransformation, IsString)

▷ AssociatedBicomplex(eta, name)

(operation)

▷ AssociatedBicomplex(eta)

(attribute)

**Returns:** a CAP natural transformation

#### 1.2.5 AsCategoryOfBicomplexes (for IsCapCategory)

▷ AsCategoryOfBicomplexes(A)

(attribute)

**Returns:** a CAP category

Return the category of bicomplexes of the Abelian category A of complexes of complexes.

#### 1.3 Attributes

#### 1.3.1 UnderlyingCategoryOfComplexesOfComplexes (for IsCapCategory)

□ UnderlyingCategoryOfComplexesOfComplexes(Bicx)

(attribute)

The category of double complexes underlying the category of bicomplexes Bicx.

#### 1.3.2 UnderlyingCapCategoryCell (for IsObject)

□ UnderlyingCapCategoryCell(B)

(attribute)

The complex of complexes underlying the bicomplex *B*.

### **Index**

Complexes

```
AsCategoryOfBicomplexes
                                                 for IsCapCategory, 9
    for IsCapCategory, 9
AssociatedBicomplex
    for IsCapNaturalTransformation, 9
    for IsCapNaturalTransformation, IsString, 9
AssociatedBicomplexFunctor
    for IsCapFunctor, 9
    for IsCapFunctor, IsString, 9
{\tt AssociatedBicomplexMorphism}
    for IsChainOrCochainMorphism, 8
AssociatedBicomplexObject
    for IsChainOrCochainComplex, 8
{\tt IsCapCategoryBicomplexCell}
    for IsCapCategoryCell andIsAttributeStor-
        ingRep, 7
IsCapCategoryBicomplexMorphism
    for IsCapCategoryBicomplexCell andIsCap-
        CategoryMorphism, 8
IsCapCategoryBicomplexObject
    for IsCapCategoryBicomplexCell andIsCap-
        CategoryObject, 7
IsCapCategoryCohomologicalBicomplex-
       Morphism
    for IsCapCategoryBicomplexMorphism, 8
IsCapCategoryCohomologicalBicomplex-
        Object
    for IsCapCategoryBicomplexObject, 8
IsCapCategoryHomologicalBicomplex-
        Morphism
    for IsCapCategoryBicomplexMorphism, 8
IsCapCategoryHomologicalBicomplex-
        Object
    for IsCapCategoryBicomplexObject, 7
UnderlyingCapCategoryCell
    for IsObject, 9
UnderlyingCategoryOfComplexesOf-
```