# Monoidal and monoidal (co)closed categories

2022.12-01

14 December 2022

**Mohamed Barakat** 

**Sebastian Gutsche** 

**Sebastian Posur** 

Tom Kuhmichel

#### **Mohamed Barakat**

Email: mohamed.barakat@uni-siegen.de

Homepage: http://algebra.mathematik.uni-siegen.de/barakat/

Address: Walter-Flex-Str. 3

57068 Siegen

#### **Sebastian Gutsche**

Germany

Email: gutsche@mathematik.uni-siegen.de

Homepage: http://algebra.mathematik.uni-siegen.de/gutsche/

Address: Department Mathematik Universität Siegen Walter-Flex-Straße 3 57068 Siegen Germany

#### **Sebastian Posur**

Email: sebastian.posur@uni-siegen.de

Homepage: http://algebra.mathematik.uni-siegen.de/posur/

Address: Department Mathematik Universität Siegen Walter-Flex-Straße 3 57068 Siegen Germany

#### **Tom Kuhmichel**

Email: tom.kuhmichel@student.uni-siegen.de

Homepage: https://github.com/TKuh

Address: Department Mathematik Universität Siegen Walter-Flex-Straße 3 57068 Siegen Germany

### **Contents**

| 1  | Mon                | oidal Categories   | 3  |
|----|--------------------|--|----|
|    | 1.1                | Monoidal Categories  | 3  |
|    | 1.2                | Additive Monoidal Categories                                       | 6  |
|    | 1.3                | Braided Monoidal Categories  | 7  |
|    | 1.4                | Symmetric Monoidal Categories                                      | 8  |
|    | 1.5                | Closed Monoidal Categories   | 8  |
|    | 1.6                | Coclosed Monoidal Categories                                       | 15 |
|    | 1.7                | Symmetric Closed Monoidal Categories                               | 22 |
|    | 1.8                | Symmetric Coclosed Monoidal Categories                             | 22 |
|    | 1.9                | Rigid Symmetric Closed Monoidal Categories                         | 22 |
|    |                    | Rigid Symmetric Coclosed Monoidal Categories                       | 24 |
|    |                    | Convenience Methods  | 27 |
|    | 1.12               | Add-methods  | 27 |
| 2  | Examples and Tests |  | 51 |
|    | 2.1                | Test functions   | 51 |
|    | 2.2                | Basics   | 55 |
| 3  | Code               | e Generation for Monodial Categories                               | 57 |
|    | 3.1                | Monoidal Categories  | 57 |
|    | 3.2                | Closed Monoidal Categories   | 57 |
|    | 3.3                | Coclosed Monoidal Categories                                       | 57 |
| 4  | The                | terminal category with multiple objects                            | 58 |
|    | 4.1                | Constructors   | 58 |
| 5  | Mon                | oidalCategories automatic generated documentation                  | 59 |
|    | 5.1                | MonoidalCategories automatic generated documentation of properties | 59 |
| ĺn | dex                |  | 61 |

### **Chapter 1**

### **Monoidal Categories**

### 1.1 Monoidal Categories

A 6-tuple  $(\mathbf{C}, \otimes, 1, \alpha, \lambda, \rho)$  consisting of

- a category C,
- a functor  $\otimes$  :  $\mathbf{C} \times \mathbf{C} \to \mathbf{C}$  compatible with the congruence of morphisms,
- an object  $1 \in \mathbb{C}$ ,
- a natural isomorphism  $\alpha_{a,b,c}$ :  $a \otimes (b \otimes c) \cong (a \otimes b) \otimes c$ ,
- a natural isomorphism  $\lambda_a : 1 \otimes a \cong a$ ,
- a natural isomorphism  $\rho_a : a \otimes 1 \cong a$ ,

is called a monoidal category, if

• for all objects a, b, c, d, the pentagon identity holds:

$$(\alpha_{a,b,c} \otimes \mathrm{id}_d) \circ \alpha_{a,b \otimes c,d} \circ (\mathrm{id}_a \otimes \alpha_{b,c,d}) \sim \alpha_{a \otimes b,c,d} \circ \alpha_{a,b,c \otimes d},$$

• for all objects a, c, the triangle identity holds:

$$(\rho_a \otimes \mathrm{id}_c) \circ \alpha_{a,1,c} \sim \mathrm{id}_a \otimes \lambda_c.$$

The corresponding GAP property is given by IsMonoidalCategory.

## 1.1.1 TensorProductOnMorphisms (for IsCapCategoryMorphism, IsCapCategory-Morphism)

▷ TensorProductOnMorphisms(alpha, beta)

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes b, a' \otimes b')$ 

The arguments are two morphisms  $\alpha: a \to a', \beta: b \to b'$ . The output is the tensor product  $\alpha \otimes \beta$ .

## 1.1.2 TensorProductOnMorphismsWithGivenTensorProducts (for IsCapCategory-Object, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategory-Object)

ho TensorProductOnMorphismsWithGivenTensorProducts(s, alpha, beta, r) (operation) **Returns:** a morphism in  $\operatorname{Hom}(a \otimes b, a' \otimes b')$ 

The arguments are an object  $s = a \otimes b$ , two morphisms  $\alpha : a \to a', \beta : b \to b'$ , and an object  $r = a' \otimes b'$ . The output is the tensor product  $\alpha \otimes \beta$ .

## 1.1.3 AssociatorRightToLeft (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ AssociatorRightToLeft(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes (b \otimes c), (a \otimes b) \otimes c)$ .

The arguments are three objects a,b,c. The output is the associator  $\alpha_{a,(b,c)}: a \otimes (b \otimes c) \to (a \otimes b) \otimes c$ .

# 1.1.4 AssociatorRightToLeftWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup AssociatorRightToLeftWithGivenTensorProducts(s, a, b, c, r) (operation) **Returns:** a morphism in Hom( $a \otimes (b \otimes c)$ ,  $(a \otimes b) \otimes c$ ).

The arguments are an object  $s=a\otimes (b\otimes c)$ , three objects a,b,c, and an object  $r=(a\otimes b)\otimes c$ . The output is the associator  $\alpha_{a,(b,c)}:a\otimes (b\otimes c)\to (a\otimes b)\otimes c$ .

### 1.1.5 AssociatorLeftToRight (for IsCapCategoryObject, IsCapCategoryObject) IsCapCategoryObject)

 $\triangleright$  AssociatorLeftToRight(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}((a \otimes b) \otimes c \rightarrow a \otimes (b \otimes c))$ .

The arguments are three objects a,b,c. The output is the associator  $\alpha_{(a,b),c}:(a\otimes b)\otimes c\to a\otimes (b\otimes c)$ .

# 1.1.6 AssociatorLeftToRightWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 ${\scriptstyle \hspace*{-0.5cm} \hspace*{-0.$ 

**Returns:** a morphism in  $\text{Hom}((a \otimes b) \otimes c \rightarrow a \otimes (b \otimes c))$ .

The arguments are an object  $s=(a\otimes b)\otimes c$ , three objects a,b,c, and an object  $r=a\otimes (b\otimes c)$ . The output is the associator  $\alpha_{(a,b),c}:(a\otimes b)\otimes c\to a\otimes (b\otimes c)$ .

#### 1.1.7 LeftUnitor (for IsCapCategoryObject)

▷ LeftUnitor(a) (attribute)

**Returns:** a morphism in  $\text{Hom}(1 \otimes a, a)$ 

The argument is an object a. The output is the left unitor  $\lambda_a : 1 \otimes a \to a$ .

### 1.1.8 LeftUnitorWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftUnitorWithGivenTensorProduct(a, s)

(operation)

**Returns:** a morphism in  $\text{Hom}(1 \otimes a, a)$ 

The arguments are an object a and an object  $s = 1 \otimes a$ . The output is the left unitor  $\lambda_a : 1 \otimes a \to a$ .

#### 1.1.9 LeftUnitorInverse (for IsCapCategoryObject)

▷ LeftUnitorInverse(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a, 1 \otimes a)$ 

The argument is an object a. The output is the inverse of the left unitor  $\lambda_a^{-1}: a \to 1 \otimes a$ .

### 1.1.10 LeftUnitorInverseWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftUnitorInverseWithGivenTensorProduct(a, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, 1 \otimes a)$ 

The argument is an object a and an object  $r = 1 \otimes a$ . The output is the inverse of the left unitor  $\lambda_a^{-1} : a \to 1 \otimes a$ .

#### 1.1.11 RightUnitor (for IsCapCategoryObject)

▷ RightUnitor(a)

(attribute)

**Returns:** a morphism in  $Hom(a \otimes 1, a)$ 

The argument is an object a. The output is the right unitor  $\rho_a : a \otimes 1 \to a$ .

### 1.1.12 RightUnitorWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ RightUnitorWithGivenTensorProduct(a, s)

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes 1, a)$ 

The arguments are an object a and an object  $s = a \otimes 1$ . The output is the right unitor  $\rho_a : a \otimes 1 \to a$ .

#### 1.1.13 RightUnitorInverse (for IsCapCategoryObject)

▷ RightUnitorInverse(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a, a \otimes 1)$ 

The argument is an object a. The output is the inverse of the right unitor  $\rho_a^{-1}: a \to a \otimes 1$ .

### 1.1.14 RightUnitorInverseWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ RightUnitorInverseWithGivenTensorProduct(a, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, a \otimes 1)$ 

The arguments are an object a and an object  $r = a \otimes 1$ . The output is the inverse of the right unitor  $\rho_a^{-1}: a \to a \otimes 1$ .

#### 1.1.15 TensorProductOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ TensorProductOnObjects(a, b)

(operation)

Returns: an object

The arguments are two objects a, b. The output is the tensor product  $a \otimes b$ .

#### 1.1.16 AddTensorProductOnObjects (for IsCapCategory, IsFunction)

▷ AddTensorProductOnObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductOnObjects.  $F:(a,b)\mapsto a\otimes b$ .

#### 1.1.17 TensorUnit (for IsCapCategory)

▷ TensorUnit(C)

(attribute)

Returns: an object

The argument is a category C. The output is the tensor unit 1 of C.

#### 1.1.18 AddTensorUnit (for IsCapCategory, IsFunction)

▷ AddTensorUnit(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorUnit.  $F: () \mapsto 1$ .

### 1.2 Additive Monoidal Categories

#### 1.2.1 LeftDistributivityExpanding (for IsCapCategoryObject, IsList)

▷ LeftDistributivityExpanding(a, L)

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes (b_1 \oplus \cdots \oplus b_n), (a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n))$ 

The arguments are an object a and a list of objects  $L = (b_1, \dots, b_n)$ . The output is the left distributivity morphism  $a \otimes (b_1 \oplus \dots \oplus b_n) \to (a \otimes b_1) \oplus \dots \oplus (a \otimes b_n)$ .

### 1.2.2 LeftDistributivityExpandingWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsList, IsCapCategoryObject)

riangleright LeftDistributivityExpandingWithGivenObjects(s, a, L, r)

(operation)

**Returns:** a morphism in Hom(s, r)

The arguments are an object  $s = a \otimes (b_1 \oplus \cdots \oplus b_n)$ , an object a, a list of objects  $L = (b_1, \ldots, b_n)$ , and an object  $r = (a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n)$ . The output is the left distributivity morphism  $s \to r$ .

#### 1.2.3 LeftDistributivityFactoring (for IsCapCategoryObject, IsList)

▷ LeftDistributivityFactoring(a, L)

(operation)

**Returns:** a morphism in  $\text{Hom}((a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n), a \otimes (b_1 \oplus \cdots \oplus b_n))$ 

The arguments are an object a and a list of objects  $L=(b_1,\ldots,b_n)$ . The output is the left distributivity morphism  $(a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n) \to a \otimes (b_1 \oplus \cdots \oplus b_n)$ .

#### 1.2.4 LeftDistributivityFactoringWithGivenObjects (for IsCapCategoryObject, Is-CapCategoryObject, IsList, IsCapCategoryObject)

 $\triangleright$  LeftDistributivityFactoringWithGivenObjects(s, a, L, r) (operation)

Returns: a morphism in Hom(s, r)

The arguments are an object  $s=(a\otimes b_1)\oplus\cdots\oplus(a\otimes b_n)$ , an object a, a list of objects  $L=(b_1,\ldots,b_n)$ , and an object  $r=a\otimes(b_1\oplus\cdots\oplus b_n)$ . The output is the left distributivity morphism  $s\to r$ .

#### 1.2.5 RightDistributivityExpanding (for IsList, IsCapCategoryObject)

⊳ RightDistributivityExpanding(L, a)

(operation)

**Returns:** a morphism in  $\text{Hom}((b_1 \oplus \cdots \oplus b_n) \otimes a, (b_1 \otimes a) \oplus \cdots \oplus (b_n \otimes a))$ 

The arguments are a list of objects  $L=(b_1,\ldots,b_n)$  and an object a. The output is the right distributivity morphism  $(b_1\oplus\cdots\oplus b_n)\otimes a\to (b_1\otimes a)\oplus\cdots\oplus (b_n\otimes a)$ .

### 1.2.6 RightDistributivityExpandingWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonupRightDistributivityExpandingWithGivenObjects(s, L, a, r) (operation)

Returns: a morphism in Hom(s, r)

The arguments are an object  $s = (b_1 \oplus \cdots \oplus b_n) \otimes a$ , a list of objects  $L = (b_1, \ldots, b_n)$ , an object a, and an object  $r = (b_1 \otimes a) \oplus \cdots \oplus (b_n \otimes a)$ . The output is the right distributivity morphism  $s \to r$ .

#### 1.2.7 RightDistributivityFactoring (for IsList, IsCapCategoryObject)

⊳ RightDistributivityFactoring(L, a)

(operation)

**Returns:** a morphism in  $\text{Hom}((b_1 \otimes a) \oplus \cdots \oplus (b_n \otimes a), (b_1 \oplus \cdots \oplus b_n) \otimes a)$ 

The arguments are a list of objects  $L = (b_1, \dots, b_n)$  and an object a. The output is the right distributivity morphism  $(b_1 \otimes a) \oplus \dots \oplus (b_n \otimes a) \to (b_1 \oplus \dots \oplus b_n) \otimes a$ .

### 1.2.8 RightDistributivityFactoringWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject, IsCapCategoryObject)

 $\verb| RightDistributivityFactoringWithGivenObjects(s, L, a, r) \\ \textbf{Returns:} \ a \ morphism \ in \ Hom(s,r) \\$ 

The arguments are an object  $s = (b_1 \otimes a) \oplus \cdots \oplus (b_n \otimes a)$ , a list of objects  $L = (b_1, \ldots, b_n)$ , an object a, and an object  $r = (b_1 \oplus \cdots \oplus b_n) \otimes a$ . The output is the right distributivity morphism  $s \to r$ .

### 1.3 Braided Monoidal Categories

A monoidal category  $\mathbb{C}$  equipped with a natural isomorphism  $B_{a,b}: a \otimes b \cong b \otimes a$  is called a *braided monoidal category* if

•  $\lambda_a \circ B_{a,1} \sim \rho_a$ ,

- $(B_{c,a} \otimes \mathrm{id}_b) \circ \alpha_{c,a,b} \circ B_{a \otimes b,c} \sim \alpha_{a,c,b} \circ (\mathrm{id}_a \otimes B_{b,c}) \circ \alpha_{a,b,c}^{-1}$
- $(\mathrm{id}_b \otimes B_{c,a}) \circ \alpha_{b,c,a}^{-1} \circ B_{a,b \otimes c} \sim \alpha_{b,a,c}^{-1} \circ (B_{a,b} \otimes \mathrm{id}_c) \circ \alpha_{a,b,c}$ .

The corresponding GAP property is given by IsBraidedMonoidalCategory.

#### 1.3.1 Braiding (for IsCapCategoryObject, IsCapCategoryObject)

▷ Braiding(a, b) (operation)

Monoidal Categories 4 1 2 1

**Returns:** a morphism in  $\text{Hom}(a \otimes b, b \otimes a)$ .

The arguments are two objects a, b. The output is the braiding  $B_{a,b}: a \otimes b \to b \otimes a$ .

### 1.3.2 BraidingWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ BraidingWithGivenTensorProducts(s, a, b, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes b, b \otimes a)$ .

The arguments are an object  $s = a \otimes b$ , two objects a, b, and an object  $r = b \otimes a$ . The output is the braiding  $B_{a,b}: a \otimes b \to b \otimes a$ .

#### 1.3.3 BraidingInverse (for IsCapCategoryObject, IsCapCategoryObject)

▷ BraidingInverse(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}(b \otimes a, a \otimes b)$ .

The arguments are two objects a,b. The output is the inverse braiding  $B_{a,b}^{-1}:b\otimes a\to a\otimes b$ .

### 1.3.4 BraidingInverseWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  BraidingInverseWithGivenTensorProducts(s, a, b, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(b \otimes a, a \otimes b)$ .

The arguments are an object  $s = b \otimes a$ , two objects a, b, and an object  $r = a \otimes b$ . The output is the inverse braiding  $B_{a,b}^{-1}: b \otimes a \to a \otimes b$ .

### 1.4 Symmetric Monoidal Categories

A braided monoidal category  $\mathbb{C}$  is called *symmetric monoidal category* if  $B_{a,b}^{-1} \sim B_{b,a}$ . The corresponding GAP property is given by IsSymmetricMonoidalCategory.

### 1.5 Closed Monoidal Categories

A monoidal category  $\mathbb{C}$  which has for each functor  $-\otimes b: \mathbb{C} \to \mathbb{C}$  a right adjoint (denoted by  $\underline{\mathrm{Hom}}(b,-)$ ) is called a *closed monoidal category*.

If no operations involving duals are installed manually, the dual objects will be derived as  $a^{\vee} := \text{Hom}(a, 1)$ .

The corresponding GAP property is called IsClosedMonoidalCategory.

#### 1.5.1 InternalHomOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ InternalHomOnObjects(a, b)

(operation)

Returns: an object

The arguments are two objects a, b. The output is the internal hom object Hom(a, b).

### 1.5.2 InternalHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ InternalHomOnMorphisms(alpha, beta)

(operation)

**Returns:** a morphism in Hom(Hom(a',b), Hom(a,b'))

The arguments are two morphisms  $\alpha: a \to a', \beta: b \to b'$ . The output is the internal hom morphism  $\operatorname{Hom}(\alpha, \beta): \operatorname{Hom}(a', b) \to \operatorname{Hom}(a, b')$ .

# 1.5.3 InternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

 $riangleright{}$  InternalHomOnMorphismsWithGivenInternalHoms(s, alpha, beta, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{A'}, b), \underline{\text{Hom}}(a, b')$ 

The arguments are an object  $s = \underline{\text{Hom}}(a',b)$ , two morphisms  $\alpha : a \to a', \beta : b \to b'$ , and an object  $r = \underline{\text{Hom}}(a,b')$ . The output is the internal hom morphism  $\underline{\text{Hom}}(\alpha,\beta) : \underline{\text{Hom}}(a',b) \to \underline{\text{Hom}}(a,b')$ .

#### 1.5.4 EvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ EvaluationMorphism(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(a,b) \otimes a,b)$ .

The arguments are two objects a,b. The output is the evaluation morphism  $\operatorname{ev}_{a,b}: \operatorname{\underline{Hom}}(a,b) \otimes a \to b$ , i.e., the counit of the tensor hom adjunction.

### 1.5.5 EvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ EvaluationMorphismWithGivenSource(a, b, s)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(a,b) \otimes a,b)$ .

The arguments are two objects a, b and an object  $s = \underline{\text{Hom}}(a, b) \otimes a$ . The output is the evaluation morphism  $\text{ev}_{a,b} : \underline{\text{Hom}}(a,b) \otimes a \to b$ , i.e., the counit of the tensor hom adjunction.

#### 1.5.6 CoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  CoevaluationMorphism(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{Hom}}(b, a \otimes b))$ .

The arguments are two objects a,b. The output is the coevaluation morphism  $coev_{a,b}: a \to \underline{Hom}(b,a\otimes b)$ , i.e., the unit of the tensor hom adjunction.

### 1.5.7 CoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject) Each CategoryObject ScapCategoryObject Each CategoryObject Each CategoryObject

 ${\scriptstyle \rhd\ Coevaluation Morphism With Given Range(a,\ b,\ r)}\\$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(a, \text{Hom}(b, a \otimes b))$ .

The arguments are two objects a, b and an object  $r = \underline{\text{Hom}}(b, a \otimes b)$ . The output is the coevaluation morphism  $\text{coev}_{a,b} : a \to \text{Hom}(b, a \otimes b)$ , i.e., the unit of the tensor hom adjunction.

### 1.5.8 TensorProductToInternalHomAdjunctionMap (for IsCapCategoryObject, IsCapCategoryMorphism)

▷ TensorProductToInternalHomAdjunctionMap(a, b, f)

(operation)

**Returns:** a morphism in Hom(a, Hom(b, c)).

The arguments are two objects a,b and a morphism  $f:a\otimes b\to c$ . The output is a morphism  $g:a\to \operatorname{\underline{Hom}}(b,c)$  corresponding to f under the tensor hom adjunction.

# 1.5.9 TensorProductToInternalHomAdjunctionMapWithGivenInternalHom (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\triangleright$  TensorProductToInternalHomAdjunctionMapWithGivenInternalHom(a, b, f, i) (operation)

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{Hom}}(b, c))$ .

The arguments are two objects a, b, a morphism  $f : a \otimes b \to c$  and an object  $i = \underline{\text{Hom}}(b, c)$ . The output is a morphism  $g : a \to \text{Hom}(b, c)$  corresponding to f under the tensor hom adjunction.

## 1.5.10 InternalHomToTensorProductAdjunctionMap (for IsCapCategoryObject, IsCapCategoryMorphism)

 ${\tt \triangleright Internal HomToTensor Product Adjunction Map(\it b, \it c, \it g)}\\$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes b, c)$ .

The arguments are two objects b, c and a morphism  $g: a \to \underline{\mathrm{Hom}}(b, c)$ . The output is a morphism  $f: a \otimes b \to c$  corresponding to g under the tensor hom adjunction.

# 1.5.11 InternalHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  InternalHomToTensorProductAdjunctionMapWithGivenTensorProduct(b, c, g, t) (operation)

**Returns:** a morphism in  $\text{Hom}(a \otimes b, c)$ .

The arguments are two objects b, c, a morphism  $g : a \to \underline{\mathrm{Hom}}(b, c)$  and an object  $t = a \otimes b$ . The output is a morphism  $f : a \otimes b \to c$  corresponding to g under the tensor hom adjunction.

### 1.5.12 MonoidalPreComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject) Object, IsCapCategoryObject)

▷ MonoidalPreComposeMorphism(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{Hom}(a,b) \otimes \text{Hom}(b,c), \text{Hom}(a,c))$ .

The arguments are three objects a,b,c. The output is the precomposition morphism Monoidal PreCompose Morphism With Given Objects a,b,c:  $\underline{\mathrm{Hom}}(a,b)\otimes\underline{\mathrm{Hom}}(b,c)\to\underline{\mathrm{Hom}}(a,c)$ .

# 1.5.13 MonoidalPreComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup MonoidalPreComposeMorphismWithGivenObjects(s, a, b, c, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(a,b) \otimes \underline{\text{Hom}}(b,c),\underline{\text{Hom}}(a,c))$ .

The arguments are an object  $s = \underline{\mathrm{Hom}}(a,b) \otimes \underline{\mathrm{Hom}}(b,c)$ , three objects a,b,c, and an object  $r = \underline{\mathrm{Hom}}(a,c)$ . The output is the precomposition morphism MonoidalPreComposeMorphismWithGivenObjects $_{a,b,c}: \underline{\mathrm{Hom}}(a,b) \otimes \underline{\mathrm{Hom}}(b,c) \to \underline{\mathrm{Hom}}(a,c)$ .

### 1.5.14 MonoidalPostComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject) Object, IsCapCategoryObject)

 $\triangleright$  MonoidalPostComposeMorphism(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{Hom}(b,c) \otimes \text{Hom}(a,b), \text{Hom}(a,c))$ .

The arguments are three objects a,b,c. The output is the postcomposition morphism MonoidalPostComposeMorphismWithGivenObjects<sub>a,b,c</sub>:  $\underline{\mathrm{Hom}}(b,c)\otimes\underline{\mathrm{Hom}}(a,b)\to\underline{\mathrm{Hom}}(a,c)$ .

# 1.5.15 MonoidalPostComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho MonoidalPostComposeMorphismWithGivenObjects(s, a, b, c, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(b,c) \otimes \underline{\text{Hom}}(a,b),\underline{\text{Hom}}(a,c))$ .

The arguments are an object  $s = \underline{\mathrm{Hom}}(b,c) \otimes \underline{\mathrm{Hom}}(a,b)$ , three objects a,b,c, and an object  $r = \underline{\mathrm{Hom}}(a,c)$ . The output is the postcomposition morphism MonoidalPostComposeMorphismWithGivenObjects<sub>a,b,c</sub>:  $\underline{\mathrm{Hom}}(b,c) \otimes \underline{\mathrm{Hom}}(a,b) \to \underline{\mathrm{Hom}}(a,c)$ .

#### 1.5.16 DualOnObjects (for IsCapCategoryObject)

▷ DualOnObjects(a)

(attribute)

Returns: an object

The argument is an object a. The output is its dual object  $a^{\vee}$ .

#### 1.5.17 DualOnMorphisms (for IsCapCategoryMorphism)

▷ DualOnMorphisms(alpha)

(attribute)

**Returns:** a morphism in  $\text{Hom}(b^{\vee}, a^{\vee})$ .

The argument is a morphism  $\alpha: a \to b$ . The output is its dual morphism  $\alpha^{\vee}: b^{\vee} \to a^{\vee}$ .

### 1.5.18 DualOnMorphismsWithGivenDuals (for IsCapCategoryObject, IsCapCategoryObject) ryMorphism, IsCapCategoryObject)

DualOnMorphismsWithGivenDuals(s, alpha, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(b^{\vee}, a^{\vee})$ .

The argument is an object  $s = b^{\vee}$ , a morphism  $\alpha : a \to b$ , and an object  $r = a^{\vee}$ . The output is the dual morphism  $\alpha^{\vee} : b^{\vee} \to a^{\vee}$ .

#### 1.5.19 EvaluationForDual (for IsCapCategoryObject)

▷ EvaluationForDual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a^{\vee} \otimes a, 1)$ .

The argument is an object a. The output is the evaluation morphism  $ev_a: a^{\vee} \otimes a \to 1$ .

### 1.5.20 EvaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ EvaluationForDualWithGivenTensorProduct(s, a, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a^{\vee} \otimes a, 1)$ .

The arguments are an object  $s = a^{\vee} \otimes a$ , an object a, and an object r = 1. The output is the evaluation morphism  $\operatorname{ev}_a : a^{\vee} \otimes a \to 1$ .

#### 1.5.21 MorphismToBidual (for IsCapCategoryObject)

▷ MorphismToBidual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a, (a^{\vee})^{\vee})$ .

The argument is an object a. The output is the morphism to the bidual  $a \to (a^{\vee})^{\vee}$ .

## 1.5.22 MorphismToBidualWithGivenBidual (for IsCapCategoryObject, IsCapCategoryObject)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, (a^{\vee})^{\vee})$ .

The arguments are an object a, and an object  $r = (a^{\vee})^{\vee}$ . The output is the morphism to the bidual  $a \to (a^{\vee})^{\vee}$ .

#### 1.5.23 TensorProductInternalHomCompatibilityMorphism (for IsList)

→ TensorProductInternalHomCompatibilityMorphism(list)

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{Hom}(a, a') \otimes \text{Hom}(b, b'), \text{Hom}(a \otimes b, a' \otimes b'))$ .

The argument is a list of four objects [a,a',b,b']. The output is the natural morphism TensorProductInternalHomCompatibilityMorphismWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\mathrm{Hom}}(a,a')\otimes \mathrm{Hom}(b,b')\to \mathrm{Hom}(a\otimes b,a'\otimes b')$ .

### 1.5.24 TensorProductInternalHomCompatibilityMorphismWithGivenObjects (for Is-CapCategoryObject, IsList, IsCapCategoryObject)

 $\triangleright$  TensorProductInternalHomCompatibilityMorphismWithGivenObjects(s, list, r) (operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(a,a') \otimes \underline{\text{Hom}}(b,b'), \underline{\text{Hom}}(a \otimes b,a' \otimes b')).$ 

The arguments are a list of four objects [a,a',b,b'], and two objects  $s=\underline{\mathrm{Hom}}(a,a')\otimes\underline{\mathrm{Hom}}(b,b')$  and  $r=\underline{\mathrm{Hom}}(a\otimes b,a'\otimes b')$ . The output is the natural morphism TensorProductInternalHomCompatibilityMorphismWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\mathrm{Hom}}(a,a')\otimes\underline{\mathrm{Hom}}(a\otimes b,a'\otimes b')$ .

#### 1.5.25 TensorProductDualityCompatibilityMorphism (for IsCapCategoryObject, Is-CapCategoryObject)

▷ TensorProductDualityCompatibilityMorphism(a, b)

**Returns:** a morphism in  $\text{Hom}(a^{\vee} \otimes b^{\vee}, (a \otimes b)^{\vee})$ .

The arguments are two objects a,b. The output is the natural morphism TensorProductDualityCompatibilityMorphismWithGivenObjects :  $a^{\vee} \otimes b^{\vee} \to (a \otimes b)^{\vee}$ .

## 1.5.26 TensorProductDualityCompatibilityMorphismWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho TensorProductDualityCompatibilityMorphismWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in  $\operatorname{Hom}(a^{\vee} \otimes b^{\vee}, (a \otimes b)^{\vee})$ .

The arguments are an object  $s = a^{\vee} \otimes b^{\vee}$ , two objects a, b, and an object  $r = (a \otimes b)^{\vee}$ . The output is the natural morphism TensorProductDualityCompatibilityMorphismWithGivenObjects<sub>a,b</sub>:  $a^{\vee} \otimes b^{\vee} \to (a \otimes b)^{\vee}$ .

### 1.5.27 MorphismFromTensorProductToInternalHom (for IsCapCategoryObject, Is-CapCategoryObject)

▷ MorphismFromTensorProductToInternalHom(a, b)

(operation)

(operation)

**Returns:** a morphism in  $\text{Hom}(a^{\vee} \otimes b, \underline{\text{Hom}}(a,b))$ .

The arguments are two objects a,b. The output is the natural morphism MorphismFromTensorProductToInternalHomWithGivenObjects $_{a,b}: a^{\vee} \otimes b \to \underline{\mathrm{Hom}}(a,b)$ .

# 1.5.28 MorphismFromTensorProductToInternalHomWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho MorphismFromTensorProductToInternalHomWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in  $\operatorname{Hom}(a^{\vee} \otimes b, \operatorname{Hom}(a,b))$ .

The arguments are an object  $s=a^\vee\otimes b$ , two objects a,b, and an object  $r=\underline{\mathrm{Hom}}(a,b)$ . The output is the natural morphism MorphismFromTensorProductToInternalHomWithGivenObjects $_{a,b}:a^\vee\otimes b\to \mathrm{Hom}(a,b)$ .

### 1.5.29 IsomorphismFromDualObjectToInternalHomIntoTensorUnit (for IsCapCategoryObject)

▷ IsomorphismFromDualObjectToInternalHomIntoTensorUnit(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a^{\vee}, \underline{\text{Hom}}(a, 1))$ .

The argument is an object a. The output is the isomorphism IsomorphismFromDualObjectToInternalHomIntoTensorUnit $_a: a^{\vee} \to \text{Hom}(a, 1)$ .

### 1.5.30 IsomorphismFromInternalHomIntoTensorUnitToDualObject (for IsCapCategoryObject)

▷ IsomorphismFromInternalHomIntoTensorUnitToDualObject(a)

(attribute)

**Returns:** a morphism in Hom(Hom $(a, 1), a^{\vee}$ ).

The argument is an object a. The output is the isomorphism IsomorphismFromInternalHomIntoTensorUnitToDualObject<sub>a</sub>:  $\underline{\text{Hom}}(a,1) \rightarrow a^{\vee}$ .

### 1.5.31 UniversalPropertyOfDual (for IsCapCategoryObject, IsCapCategoryMorphism)

▷ UniversalPropertyOfDual(t, a, alpha)

(operation)

**Returns:** a morphism in  $\text{Hom}(t, a^{\vee})$ .

The arguments are two objects t, a, and a morphism  $\alpha : t \otimes a \to 1$ . The output is the morphism  $t \to a^{\vee}$  given by the universal property of  $a^{\vee}$ .

#### 1.5.32 LambdaIntroduction (for IsCapCategoryMorphism)

▷ LambdaIntroduction(alpha)

(attribute)

**Returns:** a morphism in  $\text{Hom}(1, \underline{\text{Hom}}(a, b))$ .

The argument is a morphism  $\alpha: a \to b$ . The output is the corresponding morphism  $1 \to \underline{\mathrm{Hom}}(a,b)$  under the tensor hom adjunction.

### 1.5.33 LambdaElimination (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ LambdaElimination(a, b, alpha)

(operation)

**Returns:** a morphism in Hom(a,b).

The arguments are two objects a, b, and a morphism  $\alpha : 1 \to \underline{\text{Hom}}(a, b)$ . The output is a morphism  $a \to b$  corresponding to  $\alpha$  under the tensor hom adjunction.

#### 1.5.34 IsomorphismFromObjectToInternalHom (for IsCapCategoryObject)

▷ IsomorphismFromObjectToInternalHom(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{Hom}}(1, a))$ .

The argument is an object a. The output is the natural isomorphism  $a \to \operatorname{Hom}(1,a)$ .

### 1.5.35 IsomorphismFromObjectToInternalHomWithGivenInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

 ${\tt \triangleright} \ \, {\tt IsomorphismFromObjectToInternalHomWithGivenInternalHom(a, \ r)} \qquad \qquad {\tt (operation)}$ 

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{Hom}}(1, a))$ .

The argument is an object a, and an object  $r = \underline{\text{Hom}}(1,a)$ . The output is the natural isomorphism  $a \to \underline{\text{Hom}}(1,a)$ .

#### 1.5.36 IsomorphismFromInternalHomToObject (for IsCapCategoryObject)

▷ IsomorphismFromInternalHomToObject(a)

(attribute)

**Returns:** a morphism in Hom(Hom(1,a),a).

The argument is an object a. The output is the natural isomorphism  $\underline{\text{Hom}}(1,a) \to a$ .

### 1.5.37 IsomorphismFromInternalHomToObjectWithGivenInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromInternalHomToObjectWithGivenInternalHom(a, s) (operation) **Returns:** a morphism in Hom(Hom(1,a),a).

The argument is an object a, and an object  $s = \underline{\text{Hom}}(1,a)$ . The output is the natural isomorphism  $\text{Hom}(1,a) \to a$ .

### 1.6 Coclosed Monoidal Categories

A monoidal category  $\mathbb{C}$  which has for each functor  $-\otimes b : \mathbb{C} \to \mathbb{C}$  a left adjoint (denoted by  $\operatorname{coHom}(-,b)$ ) is called a *coclosed monoidal category*.

If no operations involving coduals are installed manually, the codual objects will be derived as  $a_{\vee} := \text{coHom}(1, a)$ .

The corresponding GAP property is called IsCoclosedMonoidalCategory.

#### 1.6.1 InternalCoHomOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ InternalCoHomOnObjects(a, b)

(operation)

Returns: an object

The arguments are two objects a, b. The output is the internal cohom object  $\underline{\operatorname{coHom}}(a, b)$ .

## 1.6.2 InternalCoHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ InternalCoHomOnMorphisms(alpha, beta)

(operation)

**Returns:** a morphism in Hom(coHom(a,b'), coHom(a',b))

The arguments are two morphisms  $\alpha: a \to a', \beta: b \to b'$ . The output is the internal cohom morphism  $\underline{\operatorname{coHom}}(\alpha, \beta): \underline{\operatorname{coHom}}(a, b') \to \underline{\operatorname{coHom}}(a', b)$ .

## 1.6.3 InternalCoHomOnMorphismsWithGivenInternalCoHoms (for IsCapCategory-Object, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategory-Object)

 $\verb| InternalCoHomOnMorphismsWithGivenInternalCoHoms(s, alpha, beta, r) | (operation) \\ \textbf{Returns:} \ a \ morphism \ in \ Hom(\underbrace{coHom}(a,b'),\underbrace{coHom}(a',b)) \\ | (operation) | (operati$ 

The arguments are an object  $s = \underline{\operatorname{coHom}}(a,b')$ , two morphisms  $\alpha : a \to a', \beta : b \to b'$ , and an object  $r = \underline{\operatorname{coHom}}(a',b)$ . The output is the internal cohom morphism  $\underline{\operatorname{coHom}}(\alpha,\beta) : \underline{\operatorname{coHom}}(a,b') \to \underline{\operatorname{coHom}}(a',b)$ .

### 1.6.4 CoclosedEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedEvaluationMorphism(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{coHom}}(a, b) \otimes b)$ .

The arguments are two objects a,b. The output is the coclosed evaluation morphism  $\operatorname{coclev}_{a,b}$ :  $a \to \operatorname{\underline{coHom}}(a,b) \otimes b$ , i.e., the unit of the cohom tensor adjunction.

### 1.6.5 CoclosedEvaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  CoclosedEvaluationMorphismWithGivenRange(a, b, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a, \text{coHom}(a, b) \otimes b)$ .

The arguments are two objects a, b and an object  $r = \underline{\operatorname{coHom}}(a, b) \otimes b$ . The output is the coclosed evaluation morphism  $\operatorname{coclev}_{a,b} : a \to \underline{\operatorname{coHom}}(a, b) \otimes b$ , i.e., the unit of the cohom tensor adjunction.

### 1.6.6 CoclosedCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  CoclosedCoevaluationMorphism(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{coHom}(a \otimes b, b), a)$ .

The arguments are two objects a,b. The output is the coclosed coevaluation morphism  $\operatorname{coclcoev}_{a,b}: \operatorname{\underline{coHom}}(a\otimes b,b)\to a$ , i.e., the counit of the cohom tensor adjunction.

### 1.6.7 CoclosedCoevaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{coHom}(a \otimes b, b), b)$ .

The arguments are two objects a,b and an object  $s = \underline{\operatorname{coHom}}(a \otimes b,b)$ . The output is the coclosed coevaluation morphism  $\operatorname{coclcoev}_{a,b} : \underline{\operatorname{coHom}}(a \otimes b,b) \to a$ , i.e., the unit of the cohom tensor adjunction.

### 1.6.8 TensorProductToInternalCoHomAdjunctionMap (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

> TensorProductToInternalCoHomAdjunctionMap(c, b, g)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,b),c)$ .

The arguments are two objects c,b and a morphism  $g:a\to c\otimes b$ . The output is a morphism  $f:\operatorname{coHom}(a,b)\to c$  corresponding to g under the cohom tensor adjunction.

# 1.6.9 TensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,b),c)$ .

The arguments are two objects c, b, a morphism  $g : a \to c \otimes b$  and an object  $i = \underline{\text{coHom}}(a, b)$ . The output is a morphism  $f : \underline{\text{coHom}}(a, b) \to c$  corresponding to g under the cohom tensor adjunction.

### 1.6.10 InternalCoHomToTensorProductAdjunctionMap (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

 $\triangleright$  InternalCoHomToTensorProductAdjunctionMap(a, b, f) (operation) **Returns:** a morphism in Hom( $a, c \otimes b$ ).

The arguments are two objects a, b and a morphism  $f : \underline{\operatorname{coHom}}(a, b) \to c$ . The output is a morphism  $g : a \to c \otimes b$  corresponding to f under the cohom tensor adjunction.

## 1.6.11 InternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject) IsCapCategoryObject)

**Returns:** a morphism in  $\text{Hom}(a, c \otimes b)$ .

The arguments are two objects a,b, a morphism  $f: \underline{\operatorname{coHom}}(a,b) \to c$  and an object  $t=c \otimes b$ . The output is a morphism  $g: a \to c \otimes b$  corresponding to f under the cohom tensor adjunction.

#### 

ightharpoonup MonoidalPreCoComposeMorphism(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,c),\underline{\text{coHom}}(b,c)\otimes\underline{\text{coHom}}(a,b))$ .

The arguments are three objects a,b,c. The output is the precocomposition morphism MonoidalPreCoComposeMorphismWithGivenObjects $_{a,b,c}: \underline{\mathrm{coHom}}(a,c) \to \underline{\mathrm{coHom}}(b,c) \otimes \underline{\mathrm{coHom}}(a,b).$ 

# 1.6.13 MonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho MonoidalPreCoComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in Hom(coHom(a,c),coHom(b,c)  $\otimes$  coHom(a,b)).

The arguments are an object  $s = \underline{\operatorname{coHom}}(a,c)$ , three objects a,b,c, and an object  $r = \underline{\operatorname{coHom}}(a,b) \otimes \underline{\operatorname{coHom}}(b,c)$ . The output is the precocomposition morphism

MonoidalPreCoComposeMorphismWithGivenObjects $_{a,b,c}$  :  $\underline{\operatorname{coHom}}(a,c) \to \underline{\operatorname{coHom}}(b,c) \otimes \underline{\operatorname{coHom}}(a,b)$ .

### 1.6.14 MonoidalPostCoComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ MonoidalPostCoComposeMorphism(a, b, c)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,c),\underline{\text{coHom}}(a,b) \otimes \underline{\text{coHom}}(b,c))$ .

The arguments are three objects a,b,c. The output is the postcocomposition morphism MonoidalPostCoComposeMorphismWithGivenObjects $_{a,b,c}$ :  $\underline{\operatorname{coHom}}(a,c) \to \underline{\operatorname{coHom}}(a,b) \otimes \underline{\operatorname{coHom}}(b,c)$ .

# 1.6.15 MonoidalPostCoComposeMorphismWithGivenObjects (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho MonoidalPostCoComposeMorphismWithGivenObjects(s, a, b, c, r)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,c),\underline{\text{coHom}}(a,b) \otimes \underline{\text{coHom}}(b,c))$ .

(operation)

The arguments are an object  $s=\underline{\operatorname{coHom}}(a,c)$ , three objects a,b,c, and an object  $r=\underline{\operatorname{coHom}}(b,c)\otimes\underline{\operatorname{coHom}}(a,b)$ . The output is the postcocomposition morphism MonoidalPostCoComposeMorphismWithGivenObjects $_{a,b,c}:\underline{\operatorname{coHom}}(a,c)\to\underline{\operatorname{coHom}}(a,b)\otimes\underline{\operatorname{coHom}}(b,c)$ .

#### 1.6.16 CoDualOnObjects (for IsCapCategoryObject)

▷ CoDualOnObjects(a)

(attribute)

Returns: an object

The argument is an object a. The output is its codual object  $a_{\lor}$ .

#### 1.6.17 CoDualOnMorphisms (for IsCapCategoryMorphism)

▷ CoDualOnMorphisms(alpha)

(attribute)

**Returns:** a morphism in  $\text{Hom}(b_{\vee}, a_{\vee})$ .

The argument is a morphism  $\alpha: a \to b$ . The output is its codual morphism  $\alpha_{\vee}: b_{\vee} \to a_{\vee}$ .

### 1.6.18 CoDualOnMorphismsWithGivenCoDuals (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

▷ CoDualOnMorphismsWithGivenCoDuals(s, alpha, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(b_{\vee}, a_{\vee})$ .

The argument is an object  $s = b_{\vee}$ , a morphism  $\alpha : a \to b$ , and an object  $r = a_{\vee}$ . The output is the dual morphism  $\alpha_{\vee} : b^{\vee} \to a^{\vee}$ .

#### 1.6.19 CoclosedEvaluationForCoDual (for IsCapCategoryObject)

▷ CoclosedEvaluationForCoDual(a)

(attribute)

**Returns:** a morphism in Hom $(1, a_{\lor} \otimes a)$ .

The argument is an object a. The output is the coclosed evaluation morphism  $\operatorname{coclev}_a: 1 \to a_{\vee} \otimes a$ .

#### 1.6.20 CoclosedEvaluationForCoDualWithGivenTensorProduct (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryObject)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$ 

**Returns:** a morphism in  $\text{Hom}(1, a_{\vee} \otimes a)$ .

The arguments are an object s=1, an object a, and an object  $r=a_{\vee}\otimes a$ . The output is the coclosed evaluation morphism  $\operatorname{coclev}_a: 1 \to a_{\vee} \otimes a$ .

#### 1.6.21 MorphismFromCoBidual (for IsCapCategoryObject)

▷ MorphismFromCoBidual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}((a_{\vee})_{\vee}, a)$ .

The argument is an object a. The output is the morphism from the cobidual  $(a_{\vee})_{\vee} \to a$ .

### 1.6.22 MorphismFromCoBidualWithGivenCoBidual (for IsCapCategoryObject, IsCapCategoryObject)

(operation)

**Returns:** a morphism in  $\text{Hom}((a_{\vee})_{\vee}, a)$ .

The arguments are an object a, and an object  $s=(a_{\vee})_{\vee}$ . The output is the morphism from the cobidual  $(a_{\vee})_{\vee} \to a$ .

#### 1.6.23 InternalCoHomTensorProductCompatibilityMorphism (for IsList)

 ${\tt \triangleright} \ \, {\tt InternalCoHomTensorProductCompatibilityMorphism}(list)\\$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a \otimes a', b \otimes b'), \underline{\text{coHom}}(a, b) \otimes \underline{\text{coHom}}(a', b'))$ .

The argument is a list of four objects [a,a',b,b']. The output is the natural morphism InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects<sub>a,a',b,b'</sub>:  $\underline{\operatorname{coHom}}(a \otimes a',b \otimes b') \to \operatorname{coHom}(a,b) \otimes \operatorname{coHom}(a',b')$ .

### 1.6.24 InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

 ${\tt \triangleright} \ \, {\tt InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects}(s, \ list, \ r) \\$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a \otimes a', b \otimes b'), \underline{\text{coHom}}(a, b) \otimes \underline{\text{coHom}}(a', b'))$ .

The arguments are a list of four objects [a,a',b,b'], and two objects  $s = \underline{\operatorname{coHom}}(a \otimes a',b \otimes b')$  and  $r = \underline{\operatorname{coHom}}(a,b) \otimes \underline{\operatorname{coHom}}(a',b')$ . The output is the natural morphism InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\operatorname{coHom}}(a \otimes a',b \otimes b') \to \operatorname{coHom}(a,b) \otimes \operatorname{coHom}(a',b')$ .

## 1.6.25 CoDualityTensorProductCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoDualityTensorProductCompatibilityMorphism(a, b)

(operation)

**Returns:** a morphism in  $\text{Hom}((a \otimes b)_{\vee}, a_{\vee} \otimes b_{\vee})$ .

The arguments are two objects a,b. The output is the natural morphism CoDualityTensorProductCompatibilityMorphismWithGivenObjects :  $(a \otimes b)_{\lor} \rightarrow a_{\lor} \otimes b_{\lor}$ .

### 1.6.26 CoDualityTensorProductCompatibilityMorphismWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  CoDualityTensorProductCompatibilityMorphismWithGivenObjects(s, a, b, r) (operation)

**Returns:** a morphism in  $\text{Hom}((a \otimes b)_{\vee}, a_{\vee} \otimes b_{\vee})$ .

The arguments are an object  $s=(a\otimes b)_\vee$ , two objects a,b, and an object  $r=a_\vee\otimes b_\vee$ . The output is the natural morphism CoDualityTensorProductCompatibilityMorphismWithGivenObjects<sub>a,b</sub>:  $(a\otimes b)_\vee\to a_\vee\otimes b_\vee$ .

### 1.6.27 MorphismFromInternalCoHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

 ${\scriptstyle \, \, \triangleright \, \, } \, \, \texttt{MorphismFromInternalCoHomToTensorProduct(a, b)}$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{coHom}}(a,b), b_{\vee} \otimes a)$ .

The arguments are two objects a,b. The output is the natural morphism MorphismFromInternalCoHomToTensorProductWithGivenObjects $_{a,b}$ :  $\underline{\operatorname{coHom}}(a,b) \to b_{\vee} \otimes a$ .

## 1.6.28 MorphismFromInternalCoHomToTensorProductWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho MorphismFromInternalCoHomToTensorProductWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in Hom(coHom(a,b), $a \otimes b_{\lor}$ ).

The arguments are an object  $s = \underline{\operatorname{coHom}}(a,b)$ , two objects a,b, and an object  $r = b_{\vee} \otimes a$ . The output is the natural morphism MorphismFromInternalCoHomToTensorProductWithGivenObjects<sub>a,b</sub>:  $\underline{\operatorname{coHom}}(a,b) \to a \otimes b_{\vee}$ .

### 1.6.29 IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit (for Is-CapCategoryObject)

▷ IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit(a) (attribute) **Returns:** a morphism in  $Hom(a_{\lor}, coHom(1, a))$ .

The argument is an object a. The output is the isomorphism IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit<sub>a</sub>:  $a \lor b$  coHom(1,a).

### 1.6.30 IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject (for Is-CapCategoryObject)

The argument is an object a. The output is the isomorphism IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject<sub>a</sub>:  $\underline{\operatorname{coHom}}(1,a) \to a_{\vee}$ .

### 1.6.31 UniversalPropertyOfCoDual (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ UniversalPropertyOfCoDual(t, a, alpha)

(operation)

**Returns:** a morphism in  $\text{Hom}(a_{\vee},t)$ .

The arguments are two objects t, a, and a morphism  $\alpha : 1 \to t \otimes a$ . The output is the morphism  $a_{\vee} \to t$  given by the universal property of  $a_{\vee}$ .

#### 1.6.32 CoLambdaIntroduction (for IsCapCategoryMorphism)

▷ CoLambdaIntroduction(alpha)

(attribute)

**Returns:** a morphism in Hom(coHom(a, b), 1).

The argument is a morphism  $\alpha : a \to b$ . The output is the corresponding morphism  $\underline{\operatorname{coHom}}(a,b) \to 1$  under the cohom tensor adjunction.

### 1.6.33 CoLambdaElimination (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ CoLambdaElimination(a, b, alpha)

(operation)

**Returns:** a morphism in Hom(a, b).

The arguments are two objects a,b, and a morphism  $\alpha : \underline{\operatorname{coHom}}(a,b) \to 1$ . The output is a morphism  $a \to b$  corresponding to  $\alpha$  under the cohom tensor adjunction.

#### 1.6.34 IsomorphismFromObjectToInternalCoHom (for IsCapCategoryObject)

 ${\tt \triangleright} \ \, {\tt IsomorphismFromObjectToInternalCoHom}(a)\\$ 

(attribute)

**Returns:** a morphism in  $\text{Hom}(a, \underline{\text{coHom}}(a, 1))$ .

The argument is an object a. The output is the natural isomorphism  $a \to \underline{\operatorname{coHom}}(a,1)$ .

### 1.6.35 IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom (for Is-CapCategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom(a, r) (operation)

Returns: a morphism in Hom(a, coHom(a, 1)).

The argument is an object a, and an object  $r = \underline{\operatorname{coHom}}(a, 1)$ . The output is the natural isomorphism  $a \to \operatorname{coHom}(a, 1)$ .

#### 1.6.36 IsomorphismFromInternalCoHomToObject (for IsCapCategoryObject)

 ${\tt \triangleright} \ \, {\tt IsomorphismFromInternalCoHomToObject(a)} \\$ 

(attribute)

**Returns:** a morphism in Hom(coHom(a, 1), a).

The argument is an object a. The output is the natural isomorphism  $\underline{\operatorname{coHom}}(a,1) \to a$ .

## 1.6.37 IsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom (for Is-CapCategoryObject, IsCapCategoryObject)

 $\triangleright$  IsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom(a, s) (operation) **Returns:** a morphism in Hom(coHom(a, 1), a).

The argument is an object a, and an object  $s = \underline{\operatorname{coHom}}(a, 1)$ . The output is the natural isomorphism  $\operatorname{coHom}(a, 1) \to a$ .

### 1.7 Symmetric Closed Monoidal Categories

A monoidal category **C** which is symmetric and closed is called a *symmetric closed monoidal category*. The corresponding GAP property is given by IsSymmetricClosedMonoidalCategory.

### 1.8 Symmetric Coclosed Monoidal Categories

A monoidal category **C** which is symmetric and coclosed is called a *symmetric coclosed monoidal* category.

The corresponding GAP property is given by IsSymmetricCoclosedMonoidalCategory.

### 1.9 Rigid Symmetric Closed Monoidal Categories

A symmetric closed monoidal category C satisfying

• the natural morphism

 $\underline{\text{Hom}}(a,a') \otimes \underline{\text{Hom}}(b,b') \to \underline{\text{Hom}}(a \otimes b,a' \otimes b')$  is an isomorphism,

- the natural morphism
- $a \to \underline{\operatorname{Hom}}(\underline{\operatorname{Hom}}(a,1),1)$  is an isomorphism is called a *rigid symmetric closed monoidal category*. If no operations involving the closed structure are installed manually, the internal hom objects will be derived as  $\underline{\operatorname{Hom}}(a,b) \coloneqq a^{\vee} \otimes b$  and, in particular,  $\underline{\operatorname{Hom}}(a,1) \coloneqq a^{\vee} \otimes 1$ .

The corresponding GAP property is given by IsRigidSymmetricClosedMonoidalCategory.

### 1.9.1 IsomorphismFromTensorProductWithDualObjectToInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromTensorProductWithDualObjectToInternalHom(a, b) (operation) **Returns:** a morphism in Hom( $a^{\vee} \otimes b$ , Hom(a, b).

The arguments are two objects a,b. The output is the natural morphism IsomorphismFromTensorProductWithDualObjectToInternalHom $_{a,b}: a^{\vee} \otimes b \to \underline{\mathrm{Hom}}(a,b)$ .

### 1.9.2 IsomorphismFromInternalHomToTensorProductWithDualObject (for IsCap-CategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromInternalHomToTensorProductWithDualObject(a, b) (operation) **Returns:** a morphism in Hom $(\underline{\text{Hom}}(a,b),a^{\vee}\otimes b)$ .

The arguments are two objects a,b. The output is the inverse of IsomorphismFromTensorProductWithDualObjectToInternalHom, namely IsomorphismFromInternalHomToTensorProductWithDualObject $_{a,b}:\underline{\mathrm{Hom}}(a,b)\to a^\vee\otimes b$ .

#### 1.9.3 MorphismFromInternalHomToTensorProduct (for IsCapCategoryObject, Is-CapCategoryObject)

 ${\tt \, \, \, \, \, \, MorphismFromInternal HomToTensorProduct(a, \, \, b) \, \, \, \, \, \, }$ 

(operation)

**Returns:** a morphism in  $\text{Hom}(\text{Hom}(a,b), a^{\vee} \otimes b)$ .

The arguments are two objects a,b. The output is the inverse of MorphismFromTensorProductToInternalHomWithGivenObjects, namely MorphismFromInternalHomToTensorProductWithGivenObjects<sub>a,b</sub>:  $\underline{\mathrm{Hom}}(a,b) \to a^{\vee} \otimes b$ .

## 1.9.4 MorphismFromInternalHomToTensorProductWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  MorphismFromInternalHomToTensorProductWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in Hom(Hom(a,b),  $a^{\lor} \otimes b$ ).

The arguments are an object  $s = \underline{\mathrm{Hom}}(a,b)$ , two objects a,b, and an object  $r = a^{\vee} \otimes b$ . The output is the inverse of MorphismFromTensorProductToInternalHomWithGivenObjects, namely MorphismFromInternalHomToTensorProductWithGivenObjects<sub>a,b</sub>:  $\underline{\mathrm{Hom}}(a,b) \to a^{\vee} \otimes b$ .

#### 1.9.5 TensorProductInternalHomCompatibilityMorphismInverse (for IsList)

→ TensorProductInternalHomCompatibilityMorphismInverse(list)

(operation)

**Returns:** a morphism in  $\text{Hom}(\underline{\text{Hom}}(a \otimes b, a' \otimes b'), \underline{\text{Hom}}(a, a') \otimes \underline{\text{Hom}}(b, b'))$ .

The argument is a list of four objects [a,a',b,b']. The output is the natural morphism TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\mathrm{Hom}}(a\otimes b,a'\otimes b')\to \underline{\mathrm{Hom}}(a,a')\otimes \underline{\mathrm{Hom}}(b,b')$ .

## 1.9.6 TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

**Returns:** a morphism in  $\text{Hom}(\text{Hom}(a \otimes b, a' \otimes b'), \text{Hom}(a, a') \otimes \text{Hom}(b, b'))$ .

The arguments are a list of four objects [a,a',b,b'], and two objects  $s = \underline{\operatorname{Hom}}(a \otimes b,a' \otimes b')$  and  $r = \underline{\operatorname{Hom}}(a,a') \otimes \underline{\operatorname{Hom}}(b,b')$ . The output is the natural morphism TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\operatorname{Hom}}(a \otimes b,a' \otimes b') \to \underline{\operatorname{Hom}}(a,a') \otimes \underline{\operatorname{Hom}}(b,b')$ .

#### 1.9.7 CoevaluationForDual (for IsCapCategoryObject)

▷ CoevaluationForDual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(1, a \otimes a^{\vee})$ .

The argument is an object a. The output is the coevaluation morphism  $coev_a: 1 \to a \otimes a^{\vee}$ .

### 1.9.8 CoevaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ CoevaluationForDualWithGivenTensorProduct(s, a, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(1, a \otimes a^{\vee})$ .

The arguments are an object s=1, an object a, and an object  $r=a\otimes a^{\vee}$ . The output is the coevaluation morphism  $\operatorname{coev}_a: 1\to a\otimes a^{\vee}$ .

#### 1.9.9 TraceMap (for IsCapCategoryMorphism)

▷ TraceMap(alpha)

(attribute)

**Returns:** a morphism in Hom(1,1).

The argument is an endomorphism  $\alpha: a \to a$ . The output is the trace morphism trace<sub> $\alpha$ </sub>:  $1 \to 1$ .

#### 1.9.10 RankMorphism (for IsCapCategoryObject)

▷ RankMorphism(a)

(attribute)

**Returns:** a morphism in Hom(1,1).

The argument is an object a. The output is the rank morphism rank<sub>a</sub>:  $1 \rightarrow 1$ .

#### 1.9.11 MorphismFromBidual (for IsCapCategoryObject)

▷ MorphismFromBidual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}((a^{\vee})^{\vee}, a)$ .

The argument is an object a. The output is the inverse of the morphism to the bidual  $(a^{\vee})^{\vee} \to a$ .

### 1.9.12 MorphismFromBidualWithGivenBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromBidualWithGivenBidual(a, s)

(operation)

**Returns:** a morphism in  $\text{Hom}((a^{\vee})^{\vee}, a)$ .

The argument is an object a, and an object  $s = (a^{\vee})^{\vee}$ . The output is the inverse of the morphism to the bidual  $(a^{\vee})^{\vee} \to a$ .

### 1.10 Rigid Symmetric Coclosed Monoidal Categories

A symmetric coclosed monoidal category C satisfying

• the natural morphism

 $\operatorname{coHom}(a \otimes a', b \otimes b') \to \operatorname{coHom}(a, b) \otimes \operatorname{coHom}(a', b')$  is an isomorphism,

• the natural morphism

 $\underline{\operatorname{coHom}}(1,\underline{\operatorname{coHom}}(1,a)) \to a$  is an isomorphism is called a *rigid symmetric coclosed monoidal cate-gory*.

If no operations involving the coclosed structure are installed manually, the internal cohom objects will be derived as  $coHom(a,b) := a \otimes b_{\vee}$  and, in particular,  $coHom(1,a) := 1 \otimes a_{\vee}$ .

The corresponding GAP property is given by IsRigidSymmetricCoclosedMonoidalCategory.

(operation)

### 1.10.1 IsomorphismFromInternalCoHomToTensorProductWithCoDualObject (for IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromInternalCoHomToTensorProductWithCoDualObject(a, b) (operation) **Returns:** a morphism in Hom( $\underline{\operatorname{coHom}}(a,b),b_{\vee}\otimes a$ ).

The arguments are two objects a,b. The output is the natural morphism IsomorphismFromInternalCoHomToTensorProductWithCoDualObjectWithGivenObjects $_{a,b}$ :  $\underline{\operatorname{coHom}}(a,b) \to b_{\vee} \otimes a$ .

### 1.10.2 IsomorphismFromTensorProductWithCoDualObjectToInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromTensorProductWithCoDualObjectToInternalCoHom(a, b) (operation) **Returns:** a morphism in Hom( $a_{\lor} \otimes b$ , coHom(b, a).

The arguments are two objects a,b. The output is the inverse of IsomorphismFromInternalCoHomToTensorProductWithCoDualObject, namely IsomorphismFromTensorProductWithCoDualObjectToInternalCoHom $_{a,b}: a_{\lor} \otimes b \to \underline{\text{coHom}}(b,a)$ .

### 1.10.3 MorphismFromTensorProductToInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject)

ho MorphismFromTensorProductToInternalCoHom(a, b) **Returns:** a morphism in  $\operatorname{Hom}(a_{\lor} \otimes b, \operatorname{\underline{coHom}}(b, a)).$ 

The arguments are two objects a,b. The output is the inverse of MorphismFromInternalCoHomToTensorProductWithGivenObjects, namely MorphismFromTensorProductToInternalCoHomWithGivenObjects $_{a,b}: a_{\vee}\otimes b \to \underline{\operatorname{coHom}}(b,a).$ 

### 1.10.4 MorphismFromTensorProductToInternalCoHomWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup MorphismFromTensorProductToInternalCoHomWithGivenObjects(s, a, b, r) (operation) Returns: a morphism in  $\operatorname{Hom}(a_{\vee} \otimes b, \operatorname{\underline{coHom}}(b, a).$ 

The arguments are an object  $s_{\vee} = a \otimes b$ , two objects a,b, and an object  $r = \underline{\operatorname{coHom}}(b,a)$ . The output is the inverse of MorphismFromInternalCoHomToTensorProductWithGivenObjects, namely MorphismFromTensorProductToInternalCoHomWithGivenObjects<sub>a,b</sub>:  $a_{\vee} \otimes b \to \underline{\operatorname{coHom}}(b,a)$ .

#### 1.10.5 InternalCoHomTensorProductCompatibilityMorphismInverse (for IsList)

 $\verb| InternalCoHomTensorProductCompatibilityMorphismInverse($list$) & (operation) \\ \textbf{Returns:} & a morphism in $Hom(\underbrace{coHom}(a,b) \otimes \underbrace{coHom}(a',b'),\underbrace{coHom}(a \otimes a',b \otimes b')$. \\ \end{aligned}$ 

The argument is a list of four objects [a,a',b,b']. The output is the natural morphism InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\operatorname{coHom}}(a,b)\otimes\underline{\operatorname{coHom}}(a',b')\to\underline{\operatorname{coHom}}(a\otimes a',b\otimes b')$ .

### 1.10.6 InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

▷ InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects(s,
list, r) (operation)

**Returns:** a morphism in  $\text{Hom}(\text{coHom}(a,b) \otimes \text{coHom}(a',b'), \text{coHom}(a \otimes a',b \otimes b').$ 

The arguments are a list of four objects [a,a',b,b'], and two objects  $s = \underline{\operatorname{coHom}}(a,b) \otimes \underline{\operatorname{coHom}}(a',b')$  and  $r = \underline{\operatorname{coHom}}(a \otimes a',b \otimes b')$ . The output is the natural morphism InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects $_{a,a',b,b'}$ :  $\underline{\operatorname{coHom}}(a,b) \otimes \underline{\operatorname{coHom}}(a',b') \to \underline{\operatorname{coHom}}(a \otimes a',b \otimes b')$ .

#### 1.10.7 CoclosedCoevaluationForCoDual (for IsCapCategoryObject)

▷ CoclosedCoevaluationForCoDual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a \otimes a_{\vee}, 1)$ .

The argument is an object a. The output is the coclosed coevaluation morphism  $\operatorname{coclcoev}_a: a \otimes a_{\vee} \to 1$ .

### 1.10.8 CoclosedCoevaluationForCoDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

 $\triangleright$  CoclosedCoevaluationForCoDualWithGivenTensorProduct(s, a, r) (operation) **Returns:** a morphism in Hom( $a \otimes a_{\lor}, 1$ ).

The arguments are an object  $s = a \otimes a_{\vee}$ , an object a, and an object r = 1. The output is the coclosed coevaluation morphism  $\operatorname{coclcoev}_a : a \otimes a_{\vee} \to 1$ .

#### 1.10.9 CoTraceMap (for IsCapCategoryMorphism)

▷ CoTraceMap(alpha)

(attribute)

**Returns:** a morphism in Hom(1,1).

The argument is an endomorphism  $\alpha: a \to a$ . The output is the cotrace morphism  $\operatorname{cotrace}_{\alpha}: 1 \to 1$ .

#### 1.10.10 CoRankMorphism (for IsCapCategoryObject)

▷ CoRankMorphism(a)

(attribute)

**Returns:** a morphism in Hom(1, 1).

The argument is an object a. The output is the corank morphism corank<sub>a</sub>:  $1 \rightarrow 1$ .

#### 1.10.11 MorphismToCoBidual (for IsCapCategoryObject)

▷ MorphismToCoBidual(a)

(attribute)

**Returns:** a morphism in  $\text{Hom}(a,(a_{\vee})_{\vee})$ .

The argument is an object a. The output is the inverse of the morphism from the cobidual  $a \to (a_{\vee})_{\vee}$ .

### 1.10.12 MorphismToCoBidualWithGivenCoBidual (for IsCapCategoryObject, Is-CapCategoryObject)

ightharpoonup MorphismToCoBidualWithGivenCoBidual(a, r)

(operation)

**Returns:** a morphism in  $\text{Hom}(a,(a_{\lor})_{\lor})$ .

The argument is an object a, and an object  $r = (a_{\vee})_{\vee}$ . The output is the inverse of the morphism from the cobidual  $a \to (a_{\vee})_{\vee}$ .

#### 1.11 Convenience Methods

#### 1.11.1 InternalHom (for IsCapCategoryCell, IsCapCategoryCell)

 $\triangleright$  InternalHom(a, b)

(operation)

Returns: a cell

This is a convenience method. The arguments are two cells a,b. The output is the internal hom cell. If a,b are two CAP objects the output is the internal Hom object  $\underline{\mathrm{Hom}}(a,b)$ . If at least one of the arguments is a CAP morphism the output is a CAP morphism, namely the internal hom on morphisms, where any object is replaced by its identity morphism.

#### 1.11.2 InternalCoHom (for IsCapCategoryCell, IsCapCategoryCell)

▷ InternalCoHom(a, b)

(operation)

Returns: a cell

This is a convenience method. The arguments are two cells a,b. The output is the internal cohom cell. If a,b are two CAP objects the output is the internal cohom object  $\underline{\operatorname{coHom}}(a,b)$ . If at least one of the arguments is a CAP morphism the output is a CAP morphism, namely the internal cohom on morphisms, where any object is replaced by its identity morphism.

#### 1.12 Add-methods

#### 1.12.1 AddLeftDistributivityExpanding (for IsCapCategory, IsFunction)

▷ AddLeftDistributivityExpanding(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftDistributivityExpanding.  $F:(a,L)\mapsto \text{LeftDistributivityExpanding}(a,L)$ .

### 1.12.2 AddLeftDistributivityExpandingWithGivenObjects (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm} \hspace*{0.5cm} \hspace*{0.5cm} \hspace*{0.5cm} \hspace*{0.5cm}} \hspace*{0.5cm} \hspace*{0.5cm}$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftDistributivityExpandingWithGivenObjects. F:  $(s,a,L,r) \mapsto \text{LeftDistributivityExpandingWithGivenObjects}(s,a,L,r)$ .

#### 1.12.3 AddLeftDistributivityFactoring (for IsCapCategory, IsFunction)

▷ AddLeftDistributivityFactoring(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftDistributivityFactoring.  $F:(a,L)\mapsto \text{LeftDistributivityFactoring}(a,L)$ .

### 1.12.4 AddLeftDistributivityFactoringWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddLeftDistributivityFactoringWithGivenObjects(C, F)

(operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftDistributivityFactoringWithGivenObjects. F:  $(s,a,L,r) \mapsto \text{LeftDistributivityFactoringWithGivenObjects}(s,a,L,r)$ .

#### 1.12.5 AddRightDistributivityExpanding (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightDistributivityExpanding.  $F:(L,a)\mapsto \text{RightDistributivityExpanding}(L,a)$ .

### 1.12.6 AddRightDistributivityExpandingWithGivenObjects (for IsCapCategory, Is-Function)

▷ AddRightDistributivityExpandingWithGivenObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightDistributivityExpandingWithGivenObjects. F:  $(s,L,a,r) \mapsto \text{RightDistributivityExpandingWithGivenObjects}(s,L,a,r)$ .

#### 1.12.7 AddRightDistributivityFactoring (for IsCapCategory, IsFunction)

▷ AddRightDistributivityFactoring(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightDistributivityFactoring.  $F:(L,a)\mapsto \text{RightDistributivityFactoring}(L,a)$ .

### 1.12.8 AddRightDistributivityFactoringWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddRightDistributivityFactoringWithGivenObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightDistributivityFactoringWithGivenObjects. F:  $(s,L,a,r) \mapsto \text{RightDistributivityFactoringWithGivenObjects}(s,L,a,r)$ .

#### 1.12.9 AddBraiding (for IsCapCategory, IsFunction)

 $\triangleright$  AddBraiding(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation Braiding.  $F:(a,b)\mapsto \text{Braiding}(a,b)$ .

#### 1.12.10 AddBraidingInverse (for IsCapCategory, IsFunction)

▷ AddBraidingInverse(C, F)

(operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation BraidingInverse.  $F:(a,b)\mapsto \text{BraidingInverse}(a,b)$ .

### 1.12.11 AddBraidingInverseWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddBraidingInverseWithGivenTensorProducts(C, F)

(operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation BraidingInverseWithGivenTensorProducts.  $F:(s,a,b,r)\mapsto BraidingInverseWithGivenTensorProducts(s,a,b,r)$ .

#### 1.12.12 AddBraidingWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddBraidingWithGivenTensorProducts(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation BraidingWithGivenTensorProducts.  $F:(s,a,b,r)\mapsto \text{BraidingWithGivenTensorProducts}(s,a,b,r).$ 

#### 1.12.13 AddCoevaluationMorphism (for IsCapCategory, IsFunction)

▷ AddCoevaluationMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoevaluationMorphism.  $F:(a,b)\mapsto \text{CoevaluationMorphism}(a,b)$ .

### 1.12.14 AddCoevaluationMorphismWithGivenRange (for IsCapCategory, IsFunction)

▷ AddCoevaluationMorphismWithGivenRange(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoevaluationMorphismWithGivenRange.  $F:(a,b,r)\mapsto$  CoevaluationMorphismWithGivenRange(a,b,r).

#### 1.12.15 AddDualOnMorphisms (for IsCapCategory, IsFunction)

▷ AddDualOnMorphisms(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation DualOnMorphisms.  $F:(alpha)\mapsto \text{DualOnMorphisms}(alpha)$ .

#### 1.12.16 AddDualOnMorphismsWithGivenDuals (for IsCapCategory, IsFunction)

▷ AddDualOnMorphismsWithGivenDuals(C, F)

(operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation DualOnMorphismsWithGivenDuals.  $F:(s,alpha,r)\mapsto$  DualOnMorphismsWithGivenDuals(s,alpha,r).

#### 1.12.17 AddDualOnObjects (for IsCapCategory, IsFunction)

▷ AddDualOnObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation DualOnObjects.  $F:(a) \mapsto \text{DualOnObjects}(a)$ .

#### 1.12.18 AddEvaluationForDual (for IsCapCategory, IsFunction)

 $\triangleright$  AddEvaluationForDual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation EvaluationForDual.  $F:(a) \mapsto \text{EvaluationForDual}(a)$ .

## 1.12.19 AddEvaluationForDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

 ${\tt \triangleright} \ \, {\tt AddEvaluationForDualWithGivenTensorProduct}(\textit{C, F})\\$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation EvaluationForDualWithGivenTensorProduct.  $F:(s,a,r)\mapsto \text{EvaluationForDualWithGivenTensorProduct}(s,a,r).$ 

#### 1.12.20 AddEvaluationMorphism (for IsCapCategory, IsFunction)

 $\triangleright$  AddEvaluationMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation EvaluationMorphism.  $F:(a,b)\mapsto \text{EvaluationMorphism}(a,b)$ .

#### 1.12.21 AddEvaluationMorphismWithGivenSource (for IsCapCategory, IsFunction)

▷ AddEvaluationMorphismWithGivenSource(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation EvaluationMorphismWithGivenSource.  $F:(a,b,s)\mapsto$  EvaluationMorphismWithGivenSource(a,b,s).

#### 1.12.22 AddInternalHomOnMorphisms (for IsCapCategory, IsFunction)

▷ AddInternalHomOnMorphisms(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalHomOnMorphisms.  $F:(al\,pha,beta)\mapsto$  InternalHomOnMorphisms $(al\,pha,beta)$ .

### 1.12.23 AddInternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategory, IsFunction)

 ${\tt \,\,\triangleright\,\,} \,\, {\tt AddInternalHomOnMorphismsWithGivenInternalHoms} \, ({\tt \it{C}}, \,\, {\tt \it{F}})$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalHomOnMorphismsWithGivenInternalHoms. F:  $(s, alpha, beta, r) \mapsto$ InternalHomOnMorphismsWithGivenInternalHoms(s, alpha, beta, r).

#### 1.12.24 AddInternalHomOnObjects (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalHomOnObjects.  $F:(a,b)\mapsto$  InternalHomOnObjects(a,b).

### 1.12.25 AddInternalHomToTensorProductAdjunctionMap (for IsCapCategory, IsFunction)

▷ AddInternalHomToTensorProductAdjunctionMap(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalHomToTensorProductAdjunctionMap.  $F:(b,c,g)\mapsto$  InternalHomToTensorProductAdjunctionMap(b,c,g).

### 1.12.26 AddInternalHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsCapCategory, IsFunction)

ightharpoonup AddInternalHomToTensorProductAdjunctionMapWithGivenTensorProduct(C, F) (operation)

**Returns:** nothing

The function This operaarguments are a category Cand F. operation tion adds the given function to the category the basic  $Internal {\tt HomToTensorProductAdjunctionMapWithGivenTensorProduct}.$  $F:(b,c,g,t)\mapsto$ InternalHomToTensorProductAdjunctionMapWithGivenTensorProduct(b, c, g, t).

### 1.12.27 AddIsomorphismFromDualObjectToInternalHomIntoTensorUnit (for IsCap-Category, IsFunction)

 $\triangleright$  AddIsomorphismFromDualObjectToInternalHomIntoTensorUnit(C, F) (operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromDualObjectToInternalHomIntoTensorUnit.  $F:(a)\mapsto \text{IsomorphismFromDualObjectToInternalHomIntoTensorUnit}(a)$ .

### 1.12.28 AddIsomorphismFromInternalHomIntoTensorUnitToDualObject (for IsCap-Category, IsFunction)

▶ AddIsomorphismFromInternalHomIntoTensorUnitToDualObject(C, F) (operation)
Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromInternalHomIntoTensorUnitToDualObject.  $F:(a)\mapsto \text{IsomorphismFromInternalHomIntoTensorUnitToDualObject}(a)$ .

### 1.12.29 AddIsomorphismFromInternalHomToObject (for IsCapCategory, IsFunction)

▷ AddIsomorphismFromInternalHomToObject(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromInternalHomToObject.  $F:(a)\mapsto$  IsomorphismFromInternalHomToObject(a).

### 1.12.30 AddIsomorphismFromInternalHomToObjectWithGivenInternalHom (for Is-CapCategory, IsFunction)

 $\verb| AddIsomorphismFromInternalHomToObjectWithGivenInternalHom($C$, $F$) \\ \textbf{Returns: } nothing$ 

The arguments F. are a category Cand function This operafunction F the operation tion adds the given to category for the basic Isomorphism From Internal Hom ToObject With Given Internal Hom. $(a,s) \mapsto$ IsomorphismFromInternalHomToObjectWithGivenInternalHom(a, s).

## ${\bf 1.12.31 \quad Add Isomorphism From Object To Internal Hom \ \ (for \ \ Is Cap Category, \ \ Is Function)}$

ightharpoonup AddIsomorphismFromObjectToInternalHom(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromObjectToInternalHom.  $F:(a)\mapsto IsomorphismFromObjectToInternalHom(a)$ .

### 1.12.32 AddIsomorphismFromObjectToInternalHomWithGivenInternalHom (for Is-CapCategory, IsFunction)

 $\triangleright$  AddIsomorphismFromObjectToInternalHomWithGivenInternalHom(C, F) (operation)

Returns: nothing

The arguments Cfunction F. are category and This operaadds the given function F the category basic operation to for the Isomorphism From Object To Internal Hom With Given Internal Hom. $(a,r) \mapsto$ IsomorphismFromObjectToInternalHomWithGivenInternalHom(a, r).

### 1.12.33 AddLambdaElimination (for IsCapCategory, IsFunction)

 $\triangleright$  AddLambdaElimination(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LambdaElimination.  $F:(a,b,alpha)\mapsto \text{LambdaElimination}(a,b,alpha).$ 

#### 1.12.34 AddLambdaIntroduction (for IsCapCategory, IsFunction)

▷ AddLambdaIntroduction(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LambdaIntroduction.  $F:(alpha)\mapsto LambdaIntroduction(alpha)$ .

#### 1.12.35 AddMonoidalPostComposeMorphism (for IsCapCategory, IsFunction)

 ${\scriptstyle \rhd} \ {\tt AddMonoidalPostComposeMorphism}({\it C}, \ {\it F})$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPostComposeMorphism.  $F:(a,b,c)\mapsto \texttt{MonoidalPostComposeMorphism}(a,b,c)$ .

### 1.12.36 AddMonoidalPostComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPostComposeMorphismWithGivenObjects. F:  $(s,a,b,c,r) \mapsto \text{MonoidalPostComposeMorphismWithGivenObjects}(s,a,b,c,r)$ .

#### 1.12.37 AddMonoidalPreComposeMorphism (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPreComposeMorphism.  $F:(a,b,c)\mapsto \texttt{MonoidalPreComposeMorphism}(a,b,c).$ 

### 1.12.38 AddMonoidalPreComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMonoidalPreComposeMorphismWithGivenObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPreComposeMorphismWithGivenObjects. F:  $(s,a,b,c,r) \mapsto \text{MonoidalPreComposeMorphismWithGivenObjects}(s,a,b,c,r)$ .

### **1.12.39** AddMorphismFromTensorProductToInternalHom (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt AddMorphismFromTensorProductToInternalHom}({\tt C, F}) \\$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromTensorProductToInternalHom.  $F:(a,b)\mapsto \texttt{MorphismFromTensorProductToInternalHom}(a,b)$ .

### 1.12.40 AddMorphismFromTensorProductToInternalHomWithGivenObjects (for Is-CapCategory, IsFunction)

▶ AddMorphismFromTensorProductToInternalHomWithGivenObjects(C, F)

(operation)

**Returns:** nothing

The arguments Cand are category function F. This operafunction F adds the given to the category for the basic operation MorphismFromTensorProductToInternalHomWithGivenObjects. F $: (s,a,b,r) \mapsto$ MorphismFromTensorProductToInternalHomWithGivenObjects(s, a, b, r).

#### 1.12.41 AddMorphismToBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismToBidual.  $F:(a) \mapsto \texttt{MorphismToBidual}(a)$ .

#### 1.12.42 AddMorphismToBidualWithGivenBidual (for IsCapCategory, IsFunction)

ightharpoonup AddMorphismToBidualWithGivenBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismToBidualWithGivenBidual.  $F:(a,r)\mapsto$ MorphismToBidualWithGivenBidual(a, r).

#### AddTensorProductDualityCompatibilityMorphism (for IsCapCategory, Is-Function)

▷ AddTensorProductDualityCompatibilityMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductDualityCompatibilityMorphism.  $F:(a,b)\mapsto$ TensorProductDualityCompatibilityMorphism(a,b).

#### AddTensorProductDualityCompatibilityMorphismWithGivenObjects (for Is-1.12.44 **CapCategory, IsFunction**)

 $\triangleright$  AddTensorProductDualityCompatibilityMorphismWithGivenObjects(C, F) (operation) Returns: nothing

arguments The category Cand function F. function F operation adds given to the category for the tion basic Tensor Product Duality Compatibility Morphism With Given Objects. $F: (s,a,b,r) \mapsto$ TensorProductDualityCompatibilityMorphismWithGivenObjects(s, a, b, r).

#### AddTensorProductInternalHomCompatibilityMorphism (for IsCapCategory, 1.12.45 **IsFunction**)

▷ AddTensorProductInternalHomCompatibilityMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductInternalHomCompatibilityMorphism. F:  $(list) \mapsto \texttt{TensorProductInternalHomCompatibilityMorphism}(list).$ 

#### Add Tensor Product Internal Hom Compatibility Morphism With Given Objects(for IsCapCategory, IsFunction)

ightharpoonup AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects(C, F) tion)

**Returns:** nothing

This The arguments Cfunction F. are category and operafunction operation tion adds given to the category the basic Tensor Product Internal Hom Compatibility Morphism With Given Objects. $(source, list, range) \mapsto \texttt{TensorProductInternalHomCompatibilityMorphismWithGivenObjects}(source, list, range$ 

### 1.12.47 AddTensorProductToInternalHomAdjunctionMap (for IsCapCategory, IsFunction)

▷ AddTensorProductToInternalHomAdjunctionMap(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductToInternalHomAdjunctionMap.  $F:(a,b,f)\mapsto TensorProductToInternalHomAdjunctionMap<math>(a,b,f)$ .

# 1.12.48 AddTensorProductToInternalHomAdjunctionMapWithGivenInternalHom (for IsCapCategory, IsFunction)

▶ AddTensorProductToInternalHomAdjunctionMapWithGivenInternalHom(C, F) (operation)
Returns: nothing

The arguments category  $\boldsymbol{C}$ F. This are a and a function operafunction F operation adds the given to the category for the basic  ${\tt TensorProductToInternalHomAdjunctionMapWithGivenInternalHom.}$  $F:(a,b,f,i)\mapsto$ TensorProductToInternalHomAdjunctionMapWithGivenInternalHom(a, b, f, i).

### 1.12.49 AddUniversalPropertyOfDual (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation UniversalPropertyOfDual.  $F:(t,a,alpha)\mapsto$  UniversalPropertyOfDual(t,a,alpha).

#### 1.12.50 AddCoDualOnMorphisms (for IsCapCategory, IsFunction)

▷ AddCoDualOnMorphisms(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoDualOnMorphisms.  $F:(al\,pha)\mapsto \texttt{CoDualOnMorphisms}(al\,pha)$ .

### 1.12.51 AddCoDualOnMorphismsWithGivenCoDuals (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} \hspace*{0.5cm} {\tt \hspace*{0.5cm}} \hspace*{0.5cm} {\tt \hspace*{0.5cm}} \hspace*{0.5cm} {\tt AddCoDual0nMorphismsWithGivenCoDuals}({\tt C, \hspace*{0.5cm}} {\tt F})$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoDualOnMorphismsWithGivenCoDuals.  $F:(s,alpha,r)\mapsto \text{CoDualOnMorphismsWithGivenCoDuals}(s,alpha,r).$ 

#### 1.12.52 AddCoDualOnObjects (for IsCapCategory, IsFunction)

▷ AddCoDualOnObjects(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoDualOnObjects.  $F:(a) \mapsto \text{CoDualOnObjects}(a)$ .

### 1.12.53 AddCoDualityTensorProductCompatibilityMorphism (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt AddCoDualityTensorProductCompatibilityMorphism}({\tt C, F}) \\$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoDualityTensorProductCompatibilityMorphism.  $F:(a,b)\mapsto \text{CoDualityTensorProductCompatibilityMorphism}(a,b)$ .

# 1.12.54 AddCoDualityTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

 $\verb| AddCoDualityTensorProductCompatibilityMorphismWithGivenObjects({\it C, F}) | (operation) \\ \textbf{Returns: } nothing$ 

The arguments are a category  $\boldsymbol{C}$ and function This operafunction the tion adds the given to category the basic operation  ${\tt CoDualityTensorProductCompatibilityMorphismWithGivenObjects}.$  $F:(s,a,b,r)\mapsto$ CoDualityTensorProductCompatibilityMorphismWithGivenObjects(s, a, b, r).

### 1.12.55 AddCoLambdaElimination (for IsCapCategory, IsFunction)

▷ AddCoLambdaElimination(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoLambdaElimination.  $F:(a,b,alpha)\mapsto \text{CoLambdaElimination}(a,b,alpha).$ 

### 1.12.56 AddCoLambdaIntroduction (for IsCapCategory, IsFunction)

ightharpoonup AddCoLambdaIntroduction(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoLambdaIntroduction.  $F:(alpha)\mapsto CoLambdaIntroduction(alpha)$ .

### 1.12.57 AddCoclosedCoevaluationMorphism (for IsCapCategory, IsFunction)

 ${\scriptstyle \rhd} \ \, {\tt AddCoclosedCoevaluationMorphism(\it C, \it F)} \\$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedCoevaluationMorphism.  $F:(a,b)\mapsto \texttt{CoclosedCoevaluationMorphism}(a,b)$ .

# 1.12.58 AddCoclosedCoevaluationMorphismWithGivenSource (for IsCapCategory, IsFunction)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedCoevaluationMorphismWithGivenSource. F:  $(a,b,s) \mapsto \texttt{CoclosedCoevaluationMorphismWithGivenSource}(a,b,s)$ .

#### 1.12.59 AddCoclosedEvaluationForCoDual (for IsCapCategory, IsFunction)

▷ AddCoclosedEvaluationForCoDual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedEvaluationForCoDual.  $F:(a)\mapsto \texttt{CoclosedEvaluationForCoDual}(a)$ .

# **1.12.60** AddCoclosedEvaluationForCoDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddCoclosedEvaluationForCoDualWithGivenTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedEvaluationForCoDualWithGivenTensorProduct.  $F:(s,a,r)\mapsto \texttt{CoclosedEvaluationForCoDualWithGivenTensorProduct}(s,a,r)$ .

#### 1.12.61 AddCoclosedEvaluationMorphism (for IsCapCategory, IsFunction)

▷ AddCoclosedEvaluationMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedEvaluationMorphism.  $F:(a,b)\mapsto \texttt{CoclosedEvaluationMorphism}(a,b)$ .

### 1.12.62 AddCoclosedEvaluationMorphismWithGivenRange (for IsCapCategory, IsFunction)

▷ AddCoclosedEvaluationMorphismWithGivenRange(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedEvaluationMorphismWithGivenRange.  $F:(a,b,r)\mapsto \texttt{CoclosedEvaluationMorphismWithGivenRange}(a,b,r).$ 

### 1.12.63 AddInternalCoHomOnMorphisms (for IsCapCategory, IsFunction)

ightharpoonup AddInternalCoHomOnMorphisms(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalCoHomOnMorphisms.  $F:(alpha,beta)\mapsto$  InternalCoHomOnMorphisms(alpha,beta).

# 1.12.64 AddInternalCoHomOnMorphismsWithGivenInternalCoHoms (for IsCap-Category, IsFunction)

▷ AddInternalCoHomOnMorphismsWithGivenInternalCoHoms(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalCoHomOnMorphismsWithGivenInternalCoHoms. F:  $(s, alpha, beta, r) \mapsto$ InternalCoHomOnMorphismsWithGivenInternalCoHoms(s, alpha, beta, r).

#### 1.12.65 AddInternalCoHomOnObjects (for IsCapCategory, IsFunction)

 $\triangleright$  AddInternalCoHomOnObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalCoHomOnObjects.  $F:(a,b)\mapsto$  InternalCoHomOnObjects(a,b).

# 1.12.66 AddInternalCoHomTensorProductCompatibilityMorphism (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} \hspace*{0.5cm} \hspace*{0.5cm}$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalCoHomTensorProductCompatibilityMorphism.  $F:(list)\mapsto \text{InternalCoHomTensorProductCompatibilityMorphism}(list).$ 

# 1.12.67 AddInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

**Returns:** nothing

function F. The arguments are category Cand This operaa the given function Fthe category the operation to for basic Internal Co Hom Tensor Product Compatibility Morphism With Given Objects.

 $(source, list, range) \mapsto \texttt{InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects}(source, list, range) \mapsto \texttt{InternalCoHomTensorProductCompatibilit$ 

### 1.12.68 AddInternalCoHomToTensorProductAdjunctionMap (for IsCapCategory, IsFunction)

▷ AddInternalCoHomToTensorProductAdjunctionMap(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation InternalCoHomToTensorProductAdjunctionMap.  $F:(a,b,f)\mapsto$  InternalCoHomToTensorProductAdjunctionMap(a,b,f).

# 1.12.69 AddInternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsCapCategory, IsFunction)

 $\triangleright$  AddInternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct(C, F) (operation)

#### **Returns:** nothing

CF. The arguments category and function This are a the given function to the category for the basic operation InternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct.  $F:(a,b,f,t)\mapsto$ InternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct(a, b, f, t).

# 1.12.70 AddIsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit (for IsCapCategory, IsFunction)

 $\verb| AddIsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit($C$, $F$) & (operation) \\ \textbf{Returns:} & nothing \\$ 

The arguments category  $\boldsymbol{C}$ are a and function This operaadds given function  $\boldsymbol{F}$ the category operation to for the basic Isomorphism From CoDual Object To Internal CoHom From Tensor Unit. $(a) \mapsto$  ${\tt IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit}(a).$ 

### 1.12.71 AddIsomorphismFromInternalCoHomFromTensorUnitToCoDualObject (for IsCapCategory, IsFunction)

 $\verb| AddIsomorphismFromInternalCoHomFromTensorUnitToCoDualObject({\it C, F}) \\ \textbf{Returns: } nothing \\ | (operation)$ 

The arguments category Care and function This operaadds function Ftion given to the category for the basic operation Isomorphism From Internal CoHom From Tensor Unit To CoDual Object.F $(a) \mapsto$  ${\tt IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject}(a).$ 

### 1.12.72 AddIsomorphismFromInternalCoHomToObject (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt AddIsomorphismFromInternalCoHomToObject}({\tt C, F}) \\$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromInternalCoHomToObject.  $F:(a)\mapsto IsomorphismFromInternalCoHomToObject(a)$ .

### 1.12.73 AddIsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom (for IsCapCategory, IsFunction)

 $\triangleright$  AddIsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom(C, F) (operation) Returns: nothing

The arguments category Cfunction are a and This operation adds given function Fto the category for the basic operation Isomorphism From Internal CoHom ToObject With Given Internal CoHom. $F:(a,s)\mapsto$ IsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom(a, s).

### **1.12.74** AddIsomorphismFromObjectToInternalCoHom (for IsCapCategory, IsFunction)

▷ AddIsomorphismFromObjectToInternalCoHom(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation IsomorphismFromObjectToInternalCoHom.  $F:(a)\mapsto IsomorphismFromObjectToInternalCoHom(a)$ .

### 1.12.75 AddIsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom (for IsCapCategory, IsFunction)

The arguments Cfunction This are category and operafunction the category the operation tion adds the given to basic Isomorphism From Object To Internal CoHom With Given Internal CoHom. $F:(a,r)\mapsto$ IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom(a, r).

#### 1.12.76 AddMonoidalPostCoComposeMorphism (for IsCapCategory, IsFunction)

ightharpoonup AddMonoidalPostCoComposeMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPostCoComposeMorphism.  $F:(a,b,c)\mapsto \texttt{MonoidalPostCoComposeMorphism}(a,b,c).$ 

# 1.12.77 AddMonoidalPostCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPostCoComposeMorphismWithGivenObjects. F:  $(s,a,b,c,r) \mapsto \texttt{MonoidalPostCoComposeMorphismWithGivenObjects}(s,a,b,c,r)$ .

### 1.12.78 AddMonoidalPreCoComposeMorphism (for IsCapCategory, IsFunction)

▷ AddMonoidalPreCoComposeMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPreCoComposeMorphism.  $F:(a,b,c)\mapsto \texttt{MonoidalPreCoComposeMorphism}(a,b,c).$ 

# 1.12.79 AddMonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMonoidalPreCoComposeMorphismWithGivenObjects(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MonoidalPreCoComposeMorphismWithGivenObjects. F:  $(s,a,b,c,r) \mapsto \texttt{MonoidalPreCoComposeMorphismWithGivenObjects}(s,a,b,c,r)$ .

#### 1.12.80 AddMorphismFromCoBidual (for IsCapCategory, IsFunction)

▷ AddMorphismFromCoBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromCoBidual.  $F:(a) \mapsto \text{MorphismFromCoBidual}(a)$ .

### 1.12.81 AddMorphismFromCoBidualWithGivenCoBidual (for IsCapCategory, IsFunction)

▷ AddMorphismFromCoBidualWithGivenCoBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromCoBidualWithGivenCoBidual.  $F:(a,s)\mapsto$  MorphismFromCoBidualWithGivenCoBidual(a,s).

### 1.12.82 AddMorphismFromInternalCoHomToTensorProduct (for IsCapCategory, IsFunction)

▷ AddMorphismFromInternalCoHomToTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromInternalCoHomToTensorProduct.  $F:(a,b)\mapsto \texttt{MorphismFromInternalCoHomToTensorProduct}(a,b).$ 

# 1.12.83 AddMorphismFromInternalCoHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

 $\triangleright$  AddMorphismFromInternalCoHomToTensorProductWithGivenObjects(C, F) (operation)

Returns: nothing

The function arguments are a category Cand F. This operation adds the given function to the category the basic operation  ${\tt MorphismFromInternalCoHomToTensorProductWithGivenObjects}.$  $F: (s,a,b,r) \mapsto$ MorphismFromInternalCoHomToTensorProductWithGivenObjects(s, a, b, r).

### 1.12.84 AddTensorProductToInternalCoHomAdjunctionMap (for IsCapCategory, Is-Function)

▷ AddTensorProductToInternalCoHomAdjunctionMap(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductToInternalCoHomAdjunctionMap.  $F:(c,b,g)\mapsto$  TensorProductToInternalCoHomAdjunctionMap(c,b,g).

# 1.12.85 AddTensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom (for IsCapCategory, IsFunction)

 $\triangleright$  AddTensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom(C, F) (operation)

**Returns:** nothing

The arguments category Cfunction are and This operaoperation the given function Fto the category for the basic TensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom.  $F:(c,b,g,i)\mapsto$ TensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom(c,b,g,i).

#### 1.12.86 AddUniversalPropertyOfCoDual (for IsCapCategory, IsFunction)

▷ AddUniversalPropertyOfCoDual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation UniversalPropertyOfCoDual.  $F:(t,a,alpha)\mapsto$  UniversalPropertyOfCoDual(t,a,alpha).

#### 1.12.87 AddAssociatorLeftToRight (for IsCapCategory, IsFunction)

▷ AddAssociatorLeftToRight(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation AssociatorLeftToRight.  $F:(a,b,c)\mapsto AssociatorLeftToRight(a,b,c).$ 

### 1.12.88 AddAssociatorLeftToRightWithGivenTensorProducts (for IsCapCategory, Is-Function)

▷ AddAssociatorLeftToRightWithGivenTensorProducts(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation AssociatorLeftToRightWithGivenTensorProducts. F:  $(s,a,b,c,r) \mapsto \text{AssociatorLeftToRightWithGivenTensorProducts}(s,a,b,c,r)$ .

### 1.12.89 AddAssociatorRightToLeft (for IsCapCategory, IsFunction)

▷ AddAssociatorRightToLeft(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation AssociatorRightToLeft.  $F:(a,b,c)\mapsto AssociatorRightToLeft(a,b,c)$ .

### 1.12.90 AddAssociatorRightToLeftWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddAssociatorRightToLeftWithGivenTensorProducts(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation AssociatorRightToLeftWithGivenTensorProducts. F:  $(s,a,b,c,r) \mapsto \text{AssociatorRightToLeftWithGivenTensorProducts}(s,a,b,c,r)$ .

### 1.12.91 AddLeftUnitor (for IsCapCategory, IsFunction)

▷ AddLeftUnitor(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftUnitor.  $F:(a) \mapsto \text{LeftUnitor}(a)$ .

#### 1.12.92 AddLeftUnitorInverse (for IsCapCategory, IsFunction)

▷ AddLeftUnitorInverse(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftUnitorInverse.  $F:(a) \mapsto \text{LeftUnitorInverse}(a)$ .

### **1.12.93** AddLeftUnitorInverseWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddLeftUnitorInverseWithGivenTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftUnitorInverseWithGivenTensorProduct.  $F:(a,r)\mapsto \text{LeftUnitorInverseWithGivenTensorProduct}(a,r).$ 

#### 1.12.94 AddLeftUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddLeftUnitorWithGivenTensorProduct(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftUnitorWithGivenTensorProduct.  $F:(a,s)\mapsto \text{LeftUnitorWithGivenTensorProduct}(a,s).$ 

### 1.12.95 AddRightUnitor (for IsCapCategory, IsFunction)

▷ AddRightUnitor(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightUnitor.  $F:(a) \mapsto \text{RightUnitor}(a)$ .

#### 1.12.96 AddRightUnitorInverse (for IsCapCategory, IsFunction)

▷ AddRightUnitorInverse(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightUnitorInverse.  $F:(a) \mapsto \text{RightUnitorInverse}(a)$ .

### 1.12.97 AddRightUnitorInverseWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddRightUnitorInverseWithGivenTensorProduct(C, F)

(operation)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightUnitorInverseWithGivenTensorProduct.  $F:(a,r)\mapsto RightUnitorInverseWithGivenTensorProduct(a,r)$ .

#### 1.12.98 AddRightUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddRightUnitorWithGivenTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RightUnitorWithGivenTensorProduct.  $F:(a,s)\mapsto \text{RightUnitorWithGivenTensorProduct}(a,s).$ 

#### 1.12.99 AddTensorProductOnMorphisms (for IsCapCategory, IsFunction)

▷ AddTensorProductOnMorphisms(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductOnMorphisms.  $F:(alpha,beta)\mapsto TensorProductOnMorphisms(alpha,beta)$ .

# **1.12.100** AddTensorProductOnMorphismsWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddTensorProductOnMorphismsWithGivenTensorProducts(C, F)

(operation)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductOnMorphismsWithGivenTensorProducts. F:  $(s, alpha, beta, r) \mapsto \text{TensorProductOnMorphismsWithGivenTensorProducts}(s, alpha, beta, r)$ .

#### 1.12.101 AddCoevaluationForDual (for IsCapCategory, IsFunction)

▷ AddCoevaluationForDual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoevaluationForDual.  $F:(a)\mapsto \texttt{CoevaluationForDual}(a)$ .

### **1.12.102** AddCoevaluationForDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ AddCoevaluationForDualWithGivenTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoevaluationForDualWithGivenTensorProduct.  $F:(s,a,r)\mapsto \text{CoevaluationForDualWithGivenTensorProduct}(s,a,r).$ 

# 1.12.103 AddIsomorphismFromInternalHomToTensorProductWithDualObject (for IsCapCategory, IsFunction)

▶ AddIsomorphismFromInternalHomToTensorProductWithDualObject(C, F) (operation)
Returns: nothing

The arguments are a category Cand function F. This operagiven tion adds the function Fto the the operation category for basic Isomorphism From Internal Hom To Tensor Product With Dual Object. $(a,b) \mapsto$ IsomorphismFromInternalHomToTensorProductWithDualObject(a,b).

# 1.12.104 AddIsomorphismFromTensorProductWithDualObjectToInternalHom (for IsCapCategory, IsFunction)

The arguments function F. This are category Cand operagiven tion adds function Fto the category for the basic operation  $(a,b) \mapsto$ Isomorphism From Tensor Product With Dual Object To Internal Hom.IsomorphismFromTensorProductWithDualObjectToInternalHom(a,b).

#### 1.12.105 AddMorphismFromBidual (for IsCapCategory, IsFunction)

 $\triangleright$  AddMorphismFromBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromBidual.  $F:(a) \mapsto \text{MorphismFromBidual}(a)$ .

### 1.12.106 AddMorphismFromBidualWithGivenBidual (for IsCapCategory, IsFunction)

 ${\tt \vartriangleright} \ \, {\tt AddMorphismFromBidualWithGivenBidual}(\textit{C, F})$ 

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromBidualWithGivenBidual.  $F:(a,s)\mapsto MorphismFromBidualWithGivenBidual(a,s)$ .

### 1.12.107 AddMorphismFromInternalHomToTensorProduct (for IsCapCategory, IsFunction)

▷ AddMorphismFromInternalHomToTensorProduct(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromInternalHomToTensorProduct.  $F:(a,b)\mapsto \texttt{MorphismFromInternalHomToTensorProduct}(a,b)$ .

# 1.12.108 AddMorphismFromInternalHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMorphismFromInternalHomToTensorProductWithGivenObjects(C, F) (operation)
Returns: nothing

The arguments are category Cand function F. This the given function Fthe category the basic operation to for  ${\tt MorphismFromInternalHomToTensorProductWithGivenObjects}.$  $: (s,a,b,r) \mapsto$ MorphismFromInternalHomToTensorProductWithGivenObjects(s, a, b, r).

### 1.12.109 AddRankMorphism (for IsCapCategory, IsFunction)

ightharpoonup AddRankMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation RankMorphism.  $F:(a)\mapsto \mathtt{RankMorphism}(a)$ .

# 1.12.110 AddTensorProductInternalHomCompatibilityMorphismInverse (for IsCap-Category, IsFunction)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductInternalHomCompatibilityMorphismInverse.  $F:(list) \mapsto \texttt{TensorProductInternalHomCompatibilityMorphismInverse}(list)$ .

# 1.12.111 AddTensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategory, IsFunction)

**Returns:** nothing

The arguments category Cand function F. This operaare a operation the given function Fto the category for the Tensor Product Internal Hom Compatibility Morphism Inverse With Given Objects. $(source, list, range) \mapsto \texttt{TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects}(source, list, range) \mapsto \texttt{TensorProductInternalHo$ 

#### 1.12.112 AddTraceMap (for IsCapCategory, IsFunction)

 $\triangleright$  AddTraceMap(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TraceMap.  $F:(alpha)\mapsto {\tt TraceMap}(alpha)$ .

### 1.12.113 AddCoRankMorphism (for IsCapCategory, IsFunction)

▷ AddCoRankMorphism(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoRankMorphism.  $F:(a) \mapsto \text{CoRankMorphism}(a)$ .

#### 1.12.114 AddCoTraceMap (for IsCapCategory, IsFunction)

▷ AddCoTraceMap(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoTraceMap.  $F:(alpha)\mapsto \text{CoTraceMap}(alpha)$ .

#### 1.12.115 AddCoclosedCoevaluationForCoDual (for IsCapCategory, IsFunction)

ightharpoonup AddCoclosedCoevaluationForCoDual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedCoevaluationForCoDual.  $F:(a)\mapsto \texttt{CoclosedCoevaluationForCoDual}(a)$ .

# 1.12.116 AddCoclosedCoevaluationForCoDualWithGivenTensorProduct (for IsCap-Category, IsFunction)

 ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$ 

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation CoclosedCoevaluationForCoDualWithGivenTensorProduct.  $F: (s, a, r) \mapsto \texttt{CoclosedCoevaluationForCoDualWithGivenTensorProduct}(s, a, r)$ .

# 1.12.117 AddInternalCoHomTensorProductCompatibilityMorphismInverse (for Is-CapCategory, IsFunction)

The arguments Cfunction F. are category and This operaadds the given function F the category operation to for the basic Internal CoHom Tensor Product Compatibility Morphism Inverse. $(list) \mapsto$  $Internal CoHom Tensor Product Compatibility Morphism Inverse (\it list).$ 

# 1.12.118 AddInternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects (for IsCapCategory, IsFunction)

#### **Returns:** nothing

The arguments category Cfunction This are and operathe given function Fto the category for the basic operation Internal CoHom Tensor Product Compatibility Morphism Inverse With Given Objects. $(source, list, range) \mapsto \texttt{InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects}(source, list, range) \mapsto \texttt{InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects}(source$ 

# 1.12.119 AddIsomorphismFromInternalCoHomToTensorProductWithCoDualObject (for IsCapCategory, IsFunction)

▶ AddIsomorphismFromInternalCoHomToTensorProductWithCoDualObject(C, F) (operation)
Returns: nothing

F. The arguments Cfunction are category and This operaoperation the given function Fthe category to for the basic Isomorphism From Internal CoHom To Tensor Product With CoDual Object. $F:(a,b)\mapsto$ IsomorphismFromInternalCoHomToTensorProductWithCoDualObject(a,b).

# 1.12.120 AddIsomorphismFromTensorProductWithCoDualObjectToInternalCoHom (for IsCapCategory, IsFunction)

▷ AddIsomorphismFromTensorProductWithCoDualObjectToInternalCoHom(C, F) (operation)
Returns: nothing

arguments F. The are category Cand function This operagiven function Fthe category the the to for basic operation Isomorphism From Tensor Product With CoDual Object To Internal CoHom. $F:(a,b)\mapsto$ Isomorphism From Tensor Product With CoDual Object To Internal CoHom (a, b).

### 1.12.121 AddMorphismFromTensorProductToInternalCoHom (for IsCapCategory, IsFunction)

▷ AddMorphismFromTensorProductToInternalCoHom(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismFromTensorProductToInternalCoHom.  $F:(a,b)\mapsto \texttt{MorphismFromTensorProductToInternalCoHom}(a,b).$ 

# 1.12.122 AddMorphismFromTensorProductToInternalCoHomWithGivenObjects (for IsCapCategory, IsFunction)

▶ AddMorphismFromTensorProductToInternalCoHomWithGivenObjects(C, F) (operation)
Returns: nothing

The arguments  $\boldsymbol{C}$ F. This are a category and a function operacategory tion adds the given function F to the for the basic operation  ${\tt MorphismFromTensorProductToInternalCoHomWithGivenObjects}.$  $F: (s,a,b,r) \mapsto$ MorphismFromTensorProductToInternalCoHomWithGivenObjects(s, a, b, r).

#### 1.12.123 AddMorphismToCoBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToCoBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismToCoBidual.  $F:(a) \mapsto \texttt{MorphismToCoBidual}(a)$ .

### 1.12.124 AddMorphismToCoBidualWithGivenCoBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToCoBidualWithGivenCoBidual(C, F)

(operation)

**Returns:** nothing

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation MorphismToCoBidualWithGivenCoBidual.  $F:(a,r)\mapsto MorphismToCoBidualWithGivenCoBidual(a,r)$ .

### **Examples and Tests**

### 2.1 Test functions

### 2.1.1 AdditiveMonoidalCategoriesTest

▷ AdditiveMonoidalCategoriesTest(cat, a, L)

(function)

The arguments are

- a CAP category cat
- an object a
- a list L of objects

This function checks for every operation declared in AdditiveMonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

#### 2.1.2 BraidedMonoidalCategoriesTest

▷ BraidedMonoidalCategoriesTest(cat, a, b)

(function)

The arguments are

- · a CAP category cat
- objects a, b

This function checks for every operation declared in BraidedMonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

### 2.1.3 ClosedMonoidalCategoriesTest

```
\triangleright ClosedMonoidalCategoriesTest(cat, a, b, c, d, alpha, beta, gamma, delta, epsilon, zeta) (function)
```

The arguments are

- a CAP category cat
- objects a, b, c, d
- a morphism  $\alpha: a \to b$
- a morphism  $\beta: c \to d$
- a morphism  $\gamma: a \otimes b \to 1$
- a morphism  $\delta : c \otimes d \to 1$
- a morphism  $\varepsilon : 1 \to \operatorname{Hom}(a,b)$
- a morphism  $\zeta: 1 \to \operatorname{Hom}(c,d)$

This function checks for every operation declared in ClosedMonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

#### 2.1.4 CoclosedMonoidalCategoriesTest

```
\triangleright CoclosedMonoidalCategoriesTest(cat, a, b, c, d, alpha, beta, gamma, delta, epsilon, zeta) (function)
```

The arguments are a CAP category *cat* objects a, b, c, d

- a morphism  $\alpha: a \to b$
- a morphism  $\beta: c \rightarrow d$
- a morphism  $\gamma: 1 \rightarrow a \otimes b$

- a morphism  $\delta: 1 \to c \otimes d$
- a morphism  $\varepsilon$  : coHom $(a,b) \to 1$
- a morphism  $\zeta$  : coHom $(c,d) \rightarrow 1$

This function checks for every operation declared in CoclosedMonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

### 2.1.5 MonoidalCategoriesTensorProductAndUnitTest

The arguments are

- a CAP category cat
- objects a, b

This function checks for every operation declared in MonoidalCategoriesTensorProductAndUnit.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

#### 2.1.6 MonoidalCategoriesTest

```
▷ MonoidalCategoriesTest(cat, a, b, c, alpha, beta) (function)
```

The arguments are

- a CAP category cat
- objects a, b, c
- a morphism  $\alpha: a \to b$
- a morphism  $\beta: c \rightarrow d$

This function checks for every operation declared in MonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

#### 2.1.7 RigidSymmetricClosedMonoidalCategoriesTest

▷ RigidSymmetricClosedMonoidalCategoriesTest(cat, a, b, c, d, alpha) (function)

The arguments are

- a CAP category cat
- objects a, b, c, d
- an endomorphism  $\alpha: a \to a$

This function checks for every object and morphism declared in RigidSymmetricClosedMonoidalCategories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

#### 2.1.8 RigidSymmetricCoclosedMonoidalCategoriesTest

▷ RigidSymmetricCoclosedMonoidalCategoriesTest(cat, a, b, c, d, alpha) (function)

The arguments are

- · a CAP category cat
- objects a, b, c, d
- an endomorphism  $\alpha: a \to a$

This function checks for every object and morphism declared in RigidSymmetricCoclosedMonoidal-Categories.gd if it is computable in the CAP category *cat*. If yes, then the operation is executed with the parameters given above and compared to the equivalent computation in the opposite category of *cat*. Pass the options

- verbose := true to output more information.
- only\_primitive\_operations := true, which is passed on to Opposite(), to only primitively install dual operations for primitively installed operations in *cat*. The advantage is, that more derivations might be tested. On the downside, this might test fewer dual\_pre/postprocessor\_funcs.

### 2.2 Basics

```
_{-} Example
gap> LoadPackage( "MonoidalCategories" );
gap> vecspaces := CreateCapCategory( "VectorSpaces" );
VectorSpaces
gap> ReadPackage( "MonoidalCategories",
         "examples/VectorSpacesMonoidalCategory.gi" );
gap> z := ZeroObject( vecspaces );
<A rational vector space of dimension 0>
gap> a := QVectorSpace( 1 );
<A rational vector space of dimension 1>
gap> b := QVectorSpace( 2 );
<A rational vector space of dimension 2>
gap> c := QVectorSpace( 3 );
<A rational vector space of dimension 3>
gap> alpha := VectorSpaceMorphism( a, [ [ 1, 0 ] ], b );
A rational vector space homomorphism with matrix:
[[1, 0]]
gap> beta := VectorSpaceMorphism( b,
                 [[1,0,0],[0,1,0]],c);
A rational vector space homomorphism with matrix:
[[1, 0, 0],
  [ 0, 1, 0 ] ]
gap> gamma := VectorSpaceMorphism( c,
                  [[0, 1, 1], [1, 0, 1], [1, 1, 0]], c);
A rational vector space homomorphism with matrix:
[[0, 1, 1],
  [ 1, 0, 1],
  [ 1, 1, 0 ] ]
gap> IsCongruentForMorphisms(
         TensorProductOnMorphisms( alpha, beta ),
         TensorProductOnMorphisms( beta, alpha ) );
false
gap> IsOne( AssociatorRightToLeft( a, b, c ) );
gap> IsCongruentForMorphisms(
         gamma, LambdaElimination( c, c, LambdaIntroduction( gamma ) ) );
gap> IsZero( TraceMap( gamma ) );
true
gap> IsCongruentForMorphisms(
          RankMorphism( DirectSum( a, b ) ), RankMorphism( c ) );
```

```
true
gap> IsOne( Braiding( b, c ) );
false
gap> IsOne( PreCompose( Braiding( b, c ), Braiding( c, b ) ) );
true
```

# Code Generation for Monodial Categories

### 3.1 Monoidal Categories

### 3.1.1 WriteFileForMonoidalStructure

This functions uses the dictionary  $key_val_rec$  to create a new monoidal structure. It generates the necessary files in the package  $package_name$  using the file-correspondence table  $files_rec$ . See the implementation for details.

### 3.2 Closed Monoidal Categories

#### 3.2.1 WriteFileForClosedMonoidalStructure

▷ WriteFileForClosedMonoidalStructure(key\_val\_rec, package\_name, files\_rec)

(function)

**Returns:** nothing

This functions uses the dictionary  $key\_val\_rec$  to create a new closed monoidal structure. It generates the necessary files in the package  $package\_name$  using the file-correspondence table  $files\_rec$ . See the implementation for details.

### 3.3 Coclosed Monoidal Categories

#### 3.3.1 WriteFileForCoclosedMonoidalStructure

(function)

**Returns:** nothing

This functions uses the dictionary key\_val\_rec to create a new coclosed monoidal structure. It generates the necessary files in the package package\_name using the file-correspondence table files\_rec. See the implementation for details.

# The terminal category with multiple objects

This is an example of a category which is created using CategoryConstructor out of no input.

This category "lies" in all doctrines and can hence be used (in conjunction with LazyCategory.

This category "lies" in all doctrines and can hence be used (in conjunction with LazyCategory) in order to check the type-correctness of the various derived methods provided by CAP or any CAP-based package.

### 4.1 Constructors

# MonoidalCategories automatic generated documentation

# 5.1 MonoidalCategories automatic generated documentation of properties

### **5.1.1** IsBraidedMonoidalCategory (for IsCapCategory)

▷ IsBraidedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being braided monoidal.

#### **5.1.2** IsClosedMonoidalCategory (for IsCapCategory)

▷ IsClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being closed monoidal.

### 5.1.3 IsCoclosedMonoidalCategory (for IsCapCategory)

▷ IsCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being coclosed monoidal.

#### **5.1.4** IsMonoidalCategory (for IsCapCategory)

▷ IsMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being monoidal.

### **5.1.5** IsStrictMonoidalCategory (for IsCapCategory)

▷ IsStrictMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being strict monoidal.

### **5.1.6** IsRigidSymmetricClosedMonoidalCategory (for IsCapCategory)

▷ IsRigidSymmetricClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being rigid symmetric closed monoidal.

#### 5.1.7 IsRigidSymmetricCoclosedMonoidalCategory (for IsCapCategory)

▷ IsRigidSymmetricCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being rigid symmetric coclosed monoidal.

### **5.1.8** IsSymmetricClosedMonoidalCategory (for IsCapCategory)

▷ IsSymmetricClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric closed monoidal.

### 5.1.9 IsSymmetricCoclosedMonoidalCategory (for IsCapCategory)

▷ IsSymmetricCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric coclosed monoidal.

### **5.1.10** IsSymmetricMonoidalCategory (for IsCapCategory)

▷ IsSymmetricMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric monoidal.

### **Index**

| AddAssociatorLeftToRight                     | for IsCapCategory, IsFunction, 38              |
|--|--|
| for IsCapCategory, IsFunction, 43            | AddCoDualityTensorProduct-                     |
| ${\tt AddAssociatorLeftToRightWithGiven-}$   | ${\tt CompatibilityMorphism}$                  |
| TensorProducts                               | for IsCapCategory, IsFunction, 37              |
| for IsCapCategory, IsFunction, 43            | AddCoDualityTensorProduct-                     |
| AddAssociatorRightToLeft                     | CompatibilityMorphismWith-                     |
| for IsCapCategory, IsFunction, 44            | GivenObjects                                   |
| AddAssociatorRightToLeftWithGiven-           | for IsCapCategory, IsFunction, 37              |
| TensorProducts                               | AddCoDualOnMorphisms                           |
| for IsCapCategory, IsFunction, 44            | for IsCapCategory, IsFunction, 36              |
| AddBraiding                                  | AddCoDualOnMorphismsWithGivenCoDuals           |
| for IsCapCategory, IsFunction, 29            | for IsCapCategory, IsFunction, 36              |
| AddBraidingInverse                           | AddCoDualOnObjects                             |
| for IsCapCategory, IsFunction, 29            | for IsCapCategory, IsFunction, 36              |
| AddBraidingInverseWithGivenTensor-           | AddCoevaluationForDual                         |
| Products                                     | for IsCapCategory, IsFunction, 46              |
| for IsCapCategory, IsFunction, 29            | ${\tt AddCoevaluationForDualWithGivenTensor-}$ |
| ${\tt AddBraidingWithGivenTensorProducts}$   | Product  |
| for IsCapCategory, IsFunction, 29            | for IsCapCategory, IsFunction, 46              |
| ${\tt AddCoclosedCoevaluationForCoDual}$     | ${\tt AddCoevaluationMorphism}$                |
| for IsCapCategory, IsFunction, 48            | for IsCapCategory, IsFunction, 29              |
| AddCoclosedCoevaluationForCoDualWith-        | ${\tt AddCoevaluationMorphismWithGivenRange}$  |
| ${\tt GivenTensorProduct}$                   | for IsCapCategory, IsFunction, 29              |
| for IsCapCategory, IsFunction, 48            | AddCoLambdaElimination                         |
| ${\tt AddCoclosedCoevaluationMorphism}$      | for IsCapCategory, IsFunction, 37              |
| for IsCapCategory, IsFunction, 37            | ${\tt AddCoLambdaIntroduction}$                |
| ${\tt AddCoclosedCoevaluationMorphismWith-}$ | for IsCapCategory, IsFunction, 37              |
| GivenSource                                  | ${\tt AddCoRankMorphism}$                      |
| for IsCapCategory, IsFunction, 38            | for IsCapCategory, IsFunction, 48              |
| ${\tt AddCoclosedEvaluationForCoDual}$       | AddCoTraceMap                                  |
| for IsCapCategory, IsFunction, 38            | for IsCapCategory, IsFunction, 48              |
| ${\tt AddCoclosedEvaluationForCoDualWith-}$  | AddDualOnMorphisms                             |
| ${\tt GivenTensorProduct}$                   | for IsCapCategory, IsFunction, 30              |
| for IsCapCategory, IsFunction, 38            | ${\tt AddDualOnMorphismsWithGivenDuals}$       |
| ${\tt AddCoclosedEvaluationMorphism}$        | for IsCapCategory, IsFunction, 30              |
| for IsCapCategory, IsFunction, 38            | AddDualOnObjects                               |
| ${\tt AddCoclosedEvaluationMorphismWith-}$   | for IsCapCategory, IsFunction, 30              |
| GivenRange                                   | AddEvaluationForDual                           |

| for IsCapCategory, IsFunction, 30     | ${\tt AddInternalHomToTensorProduct-}$          |
|---------------------------------------|---|
| AddEvaluationForDualWithGivenTensor-  | ${\tt AdjunctionMapWithGivenTensor-}$           |
| Product                               | Product   |
| for IsCapCategory, IsFunction, 30     | for IsCapCategory, IsFunction, 31               |
| AddEvaluationMorphism                 | AddIsomorphismFromCoDualObjectTo-               |
| for IsCapCategory, IsFunction, 30     | InternalCoHomFromTensorUnit                     |
| AddEvaluationMorphismWithGivenSource  | for IsCapCategory, IsFunction, 40               |
| for IsCapCategory, IsFunction, 31     | AddIsomorphismFromDualObjectTo-                 |
| AddInternalCoHomOnMorphisms           | $\overline{\mathtt{InternalHomIntoTensorUnit}}$ |
| for IsCapCategory, IsFunction, 38     | for IsCapCategory, IsFunction, 32               |
| AddInternalCoHomOnMorphismsWithGiven- | AddIsomorphismFromInternalCoHomFrom-            |
| InternalCoHoms                        | TensorUnitToCoDualObject                        |
| for IsCapCategory, IsFunction, 39     | for IsCapCategory, IsFunction, 40               |
| AddInternalCoHomOnObjects             | AddIsomorphismFromInternalCoHomTo-              |
| for IsCapCategory, IsFunction, 39     | Object  |
| AddInternalCoHomTensorProduct-        | for IsCapCategory, IsFunction, 40               |
| ${\tt CompatibilityMorphism}$         | AddIsomorphismFromInternalCoHomTo-              |
| for IsCapCategory, IsFunction, 39     | ObjectWithGivenInternalCoHom                    |
| AddInternalCoHomTensorProduct-        | for IsCapCategory, IsFunction, 41               |
| ${	t Compatibility Morphism Inverse}$ | AddIsomorphismFromInternalCoHomTo-              |
| for IsCapCategory, IsFunction, 49     | TensorProductWithCoDualObject                   |
| AddInternalCoHomTensorProduct-        | for IsCapCategory, IsFunction, 49               |
| CompatibilityMorphismInverse-         | AddIsomorphismFromInternalHomInto-              |
| WithGivenObjects                      | TensorUnitToDualObject                          |
| for IsCapCategory, IsFunction, 49     | for IsCapCategory, IsFunction, 32               |
| AddInternalCoHomTensorProduct-        | AddIsomorphismFromInternalHomToObject           |
| CompatibilityMorphismWith-            | for IsCapCategory, IsFunction, 32               |
| GivenObjects                          | AddIsomorphismFromInternalHomToObject           |
| for IsCapCategory, IsFunction, 39     | WithGivenInternalHom                            |
| AddInternalCoHomToTensorProduct-      | for IsCapCategory, IsFunction, 32               |
| AdjunctionMap                         | ${\tt AddIsomorphismFromInternalHomToTensor}$   |
| for IsCapCategory, IsFunction, 39     | ProductWithDualObject                           |
| AddInternalCoHomToTensorProduct-      | for IsCapCategory, IsFunction, 46               |
| ${\tt AdjunctionMapWithGivenTensor-}$ | ${\tt AddIsomorphismFromObjectToInternal-}$     |
| Product                               | CoHom   |
| for IsCapCategory, IsFunction, 40     | for IsCapCategory, IsFunction, 41               |
| AddInternalHomOnMorphisms             | ${\tt AddIsomorphismFromObjectToInternalCo-}$   |
| for IsCapCategory, IsFunction, 31     | ${\tt HomWithGivenInternalCoHom}$               |
| AddInternalHomOnMorphismsWithGiven-   | for IsCapCategory, IsFunction, 41               |
| InternalHoms                          | ${\tt AddIsomorphismFromObjectToInternalHom}$   |
| for IsCapCategory, IsFunction, 31     | for IsCapCategory, IsFunction, 32               |
| AddInternalHomOnObjects               | ${\tt AddIsomorphismFromObjectToInternalHom}$   |
| for IsCapCategory, IsFunction, 31     | ${\tt WithGivenInternalHom}$                    |
| AddInternalHomToTensorProduct-        | for IsCapCategory, IsFunction, 33               |
| ${\tt AdjunctionMap}$                 | ${\tt AddIsomorphismFromTensorProductWithCo}$   |
| for IsCapCategory, IsFunction, 31     | ${	t DualObjectToInternalCoHom}$                |

| for IsCapCategory, IsFunction, 49            | GivenObjects                                 |
|--|--|
| ${\tt AddIsomorphismFromTensorProductWith-}$ | for IsCapCategory, IsFunction, 34            |
| ${	t DualObjectToInternalHom}$               | ${\tt AddMorphismFromBidual}$                |
| for IsCapCategory, IsFunction, 46            | for IsCapCategory, IsFunction, 46            |
| AdditiveMonoidalCategoriesTest, 51           | ${\tt AddMorphismFromBidualWithGivenBidual}$ |
| AddLambdaElimination                         | for IsCapCategory, IsFunction, 47            |
| for IsCapCategory, IsFunction, 33            | AddMorphismFromCoBidual                      |
| AddLambdaIntroduction                        | for IsCapCategory, IsFunction, 42            |
| for IsCapCategory, IsFunction, 33            | AddMorphismFromCoBidualWithGivenCo-          |
| AddLeftDistributivityExpanding               | Bidual                                       |
| for IsCapCategory, IsFunction, 27            | for IsCapCategory, IsFunction, 42            |
| AddLeftDistributivityExpandingWith-          | AddMorphismFromInternalCoHomToTensor-        |
| GivenObjects                                 | Product                                      |
| for IsCapCategory, IsFunction, 27            | for IsCapCategory, IsFunction, 42            |
| AddLeftDistributivityFactoring               | AddMorphismFromInternalCoHomToTensor-        |
| for IsCapCategory, IsFunction, 28            | ProductWithGivenObjects                      |
| AddLeftDistributivityFactoringWith-          | for IsCapCategory, IsFunction, 42            |
| GivenObjects                                 | AddMorphismFromInternalHomToTensor-          |
| for IsCapCategory, IsFunction, 28            | Product                                      |
| AddLeftUnitor                                | for IsCapCategory, IsFunction, 47            |
| for IsCapCategory, IsFunction, 44            | AddMorphismFromInternalHomToTensor-          |
| AddLeftUnitorInverse                         | ProductWithGivenObjects                      |
| for IsCapCategory, IsFunction, 44            | for IsCapCategory, IsFunction, 47            |
| AddLeftUnitorInverseWithGivenTensor-         | AddMorphismFromTensorProductTo-              |
| Product                                      | InternalCoHom                                |
| for IsCapCategory, IsFunction, 44            | for IsCapCategory, IsFunction, 50            |
| AddLeftUnitorWithGivenTensorProduct          | AddMorphismFromTensorProductTo-              |
| for IsCapCategory, IsFunction, 44            | InternalCoHomWithGivenObjects                |
| AddMonoidalPostCoComposeMorphism             | for IsCapCategory, IsFunction, 50            |
| for IsCapCategory, IsFunction, 41            | AddMorphismFromTensorProductTo-              |
| AddMonoidalPostCoComposeMorphismWith-        | InternalHom                                  |
| GivenObjects                                 | for IsCapCategory, IsFunction, 34            |
| for IsCapCategory, IsFunction, 41            | AddMorphismFromTensorProductTo-              |
| AddMonoidalPostComposeMorphism               | InternalHomWithGivenObjects                  |
| for IsCapCategory, IsFunction, 33            | for IsCapCategory, IsFunction, 34            |
| AddMonoidalPostComposeMorphismWith-          | AddMorphismToBidual                          |
| GivenObjects                                 | for IsCapCategory, IsFunction, 34            |
| for IsCapCategory, IsFunction, 33            | AddMorphismToBidualWithGivenBidual           |
| AddMonoidalPreCoComposeMorphism              | for IsCapCategory, IsFunction, 34            |
| for IsCapCategory, IsFunction, 42            | AddMorphismToCoBidual                        |
| AddMonoidalPreCoComposeMorphismWith-         | for IsCapCategory, IsFunction, 50            |
| GivenObjects                                 | AddMorphismToCoBidualWithGivenCoBidual       |
| for IsCapCategory, IsFunction, 42            | for IsCapCategory, IsFunction, 50            |
| AddMonoidalPreComposeMorphism                | AddRankMorphism                              |
| for IsCapCategory, IsFunction, 34            | for IsCapCategory, IsFunction, 47            |
| AddMonoidalPreComposeMorphismWith-           | AddRightDistributivityExpanding              |

| for IsCapCategory, IsFunction, 28     | ${\tt AdjunctionMap}$                                |
|---------------------------------------|--|
| AddRightDistributivityExpandingWith-  | for IsCapCategory, IsFunction, 43                    |
| GivenObjects                          | AddTensorProductToInternalCoHom-                     |
| for IsCapCategory, IsFunction, 28     | ${\tt AdjunctionMapWithGivenInternal-}$              |
| AddRightDistributivityFactoring       | CoHom  |
| for IsCapCategory, IsFunction, 28     | for IsCapCategory, IsFunction, 43                    |
| AddRightDistributivityFactoringWith-  | AddTensorProductToInternalHom-                       |
| GivenObjects                          | AdjunctionMap  |
| for IsCapCategory, IsFunction, 28     | for IsCapCategory, IsFunction, 36                    |
| AddRightUnitor                        | AddTensorProductToInternalHom-                       |
| for IsCapCategory, IsFunction, 45     | AdjunctionMapWithGiven-                              |
| AddRightUnitorInverse                 | InternalHom  |
| for IsCapCategory, IsFunction, 45     | for IsCapCategory, IsFunction, 36                    |
| AddRightUnitorInverseWithGivenTensor- | AddTensorUnit  |
| Product                               | for IsCapCategory, IsFunction, 6                     |
| for IsCapCategory, IsFunction, 45     | AddTraceMap  |
| AddRightUnitorWithGivenTensorProduct  | for IsCapCategory, IsFunction, 48                    |
| for IsCapCategory, IsFunction, 45     | AddUniversalPropertyOfCoDual                         |
| AddTensorProductDualityCompatibility- | for IsCapCategory, IsFunction, 43                    |
| Morphism                              | AddUniversalPropertyOfDual                           |
| for IsCapCategory, IsFunction, 35     | for IsCapCategory, IsFunction, 36                    |
| AddTensorProductDualityCompatibility- | AssociatorLeftToRight                                |
| MorphismWithGivenObjects              | for IsCapCategoryObject, IsCapCategory-              |
| for IsCapCategory, IsFunction, 35     | Object, IsCapCategoryObject, 4                       |
| AddTensorProductInternalHom-          | AssociatorLeftToRightWithGivenTensor-                |
| CompatibilityMorphism                 | Products   |
| for IsCapCategory, IsFunction, 35     | for IsCapCategoryObject, IsCapCategory-              |
| AddTensorProductInternalHom-          | Object, IsCapCategoryObject, IsCap-                  |
| CompatibilityMorphismInverse          | CategoryObject, IsCapCategoryObject,                 |
| for IsCapCategory, IsFunction, 47     | 4  |
| AddTensorProductInternalHom-          | AssociatorRightToLeft                                |
| CompatibilityMorphismInverse-         | for IsCapCategoryObject, IsCapCategory-              |
| WithGivenObjects                      | Object, IsCapCategoryObject, 4                       |
| for IsCapCategory, IsFunction, 48     | AssociatorRightToLeftWithGivenTensor-                |
| AddTensorProductInternalHom-          | Products   |
| CompatibilityMorphismWith-            | for IsCapCategoryObject, IsCapCategory-              |
| GivenObjects                          | Object, IsCapCategoryObject, IsCap-                  |
| for IsCapCategory, IsFunction, 35     | CategoryObject, IsCapCategoryObject,                 |
| AddTensorProductOnMorphisms           | 4  |
| for IsCapCategory, IsFunction, 45     | ·  |
| AddTensorProductOnMorphismsWithGiven- | BraidedMonoidalCategoriesTest, 51                    |
| TensorProducts                        | Braiding   |
| for IsCapCategory, IsFunction, 45     | for IsCapCategoryObject, IsCapCategory-              |
| AddTensorProductOnObjects             | Object, 8  |
| for IsCapCategory, IsFunction, 6      | BraidingInverse                                      |
| AddTensorProductToInternalCoHom-      | for IsCapCategoryObject, IsCapCategory-<br>Object, 8 |

| ${\tt BraidingInverseWithGivenTensorProducts}$ | for IsCapCategoryMorphism, 18           |
|--|---|
| for IsCapCategoryObject, IsCapCategory-        | CoDualOnMorphismsWithGivenCoDuals       |
| Object, IsCapCategoryObject, IsCap-            | for IsCapCategoryObject, IsCapCategory- |
| CategoryObject, 8                              | Morphism, IsCapCategoryObject, 18       |
| BraidingWithGivenTensorProducts                | CoDualOnObjects                         |
| for IsCapCategoryObject, IsCapCategory-        | for IsCapCategoryObject, 18             |
| Object, IsCapCategoryObject, IsCap-            | CoevaluationForDual                     |
| CategoryObject, 8                              | for IsCapCategoryObject, 23             |
|  | CoevaluationForDualWithGivenTensor-     |
| ClosedMonoidalCategoriesTest, 52               | Product                                 |
| ${\tt CoclosedCoevaluationForCoDual}$          | for IsCapCategoryObject, IsCapCategory- |
| for IsCapCategoryObject, 26                    | Object, IsCapCategoryObject, 24         |
| ${\tt CoclosedCoevaluationForCoDualWith-}$     | CoevaluationMorphism                    |
| ${\tt GivenTensorProduct}$                     | for IsCapCategoryObject, IsCapCategory- |
| for IsCapCategoryObject, IsCapCategory-        | Object, 9                               |
| Object, IsCapCategoryObject, 26                | CoevaluationMorphismWithGivenRange      |
| ${\tt CoclosedCoevaluationMorphism}$           | for IsCapCategoryObject, IsCapCategory- |
| for IsCapCategoryObject, IsCapCategory-        | Object, IsCapCategoryObject, 10         |
| Object, 16                                     | CoLambdaElimination                     |
| CoclosedCoevaluationMorphismWithGiven-         | for IsCapCategoryObject, IsCapCategory- |
| Source   | Object, IsCapCategoryMorphism, 21       |
| for IsCapCategoryObject, IsCapCategory-        | CoLambdaIntroduction                    |
| Object, IsCapCategoryObject, 16                | for IsCapCategoryMorphism, 21           |
| CoclosedEvaluationForCoDual                    | CoRankMorphism                          |
| for IsCapCategoryObject, 18                    | for IsCapCategoryObject, 26             |
| CoclosedEvaluationForCoDualWithGiven-          | CoTraceMap                              |
| TensorProduct                                  | for IsCapCategoryMorphism, 26           |
| for IsCapCategoryObject, IsCapCategory-        | for iscapcategoryworphism, 20           |
| Object, IsCapCategoryObject, 19                | DualOnMorphisms                         |
| CoclosedEvaluationMorphism                     | for IsCapCategoryMorphism, 11           |
| for IsCapCategoryObject, IsCapCategory-        | DualOnMorphismsWithGivenDuals           |
| Object, 16                                     | for IsCapCategoryObject, IsCapCategory- |
| CoclosedEvaluationMorphismWithGiven-           | Morphism, IsCapCategoryObject, 12       |
| Range  | DualOnObjects                           |
| for IsCapCategoryObject, IsCapCategory-        | for IsCapCategoryObject, 11             |
| Object, IsCapCategoryObject, 16                |   |
| CoclosedMonoidalCategoriesTest, 52             | EvaluationForDual                       |
| CoDualityTensorProductCompatibility-           | for IsCapCategoryObject, 12             |
| Morphism                                       | EvaluationForDualWithGivenTensor-       |
| for IsCapCategoryObject, IsCapCategory-        | Product                                 |
| Object, 19                                     | for IsCapCategoryObject, IsCapCategory- |
| CoDualityTensorProductCompatibility-           | Object, IsCapCategoryObject, 12         |
| MorphismWithGivenObjects                       | EvaluationMorphism                      |
| for IsCapCategoryObject, IsCapCategory-        | for IsCapCategoryObject, IsCapCategory- |
| Object, IsCapCategoryObject, IsCap-            | Object, 9                               |
| CategoryObject, 20                             | EvaluationMorphismWithGivenSource       |
| CoDualOnMorphisms                              | <del>-</del>                            |
| 002 aaromior piiromo                           |   |

| for IsCapCategoryObject, IsCapCategory-                          | for IsCapCategoryMorphism, IsCapCatego-        |
|--|--|
| Object, IsCapCategoryObject, 9                                   | ryMorphism, 9                                  |
|  | InternalHomOnMorphismsWithGiven-               |
| InternalCoHom  | InternalHoms                                   |
| for IsCapCategoryCell, IsCapCategoryCell, 27                     | for IsCapCategoryObject, IsCapCategory-        |
|  | Morphism, IsCapCategoryMorphism                |
| InternalCoHomOnMorphisms  for IsConCotogonyMorphisms IsConCotogo | IsCapCategoryObject, 9                         |
| for IsCapCategoryMorphism, IsCapCatego-                          | InternalHomOnObjects                           |
| ryMorphism, 15   | for IsCapCategoryObject, IsCapCategory-        |
| InternalCoHomOnMorphismsWithGiven-                               | Object, 9                                      |
| InternalCoHoms   | InternalHomToTensorProduct-                    |
| for IsCapCategoryObject, IsCapCategory-                          | ${\tt AdjunctionMap}$                          |
| Morphism, IsCapCategoryMorphism,                                 | for IsCapCategoryObject, IsCapCategory-        |
| IsCapCategoryObject, 16  | Object, IsCapCategoryMorphism, 10              |
| InternalCoHomOnObjects   | $Internal {\tt HomToTensorProductAdjunction-}$ |
| for IsCapCategoryObject, IsCapCategory-                          | ${\tt MapWithGivenTensorProduct}$              |
| Object, 15   | for IsCapCategoryObject, IsCapCategory-        |
| InternalCoHomTensorProduct-                                      | Object, IsCapCategoryMorphism, Is-             |
| CompatibilityMorphism  | CapCategoryObject, 10                          |
| for IsList, 19   | IsBraidedMonoidalCategory                      |
| InternalCoHomTensorProduct-                                      | for IsCapCategory, 59                          |
| ${\tt CompatibilityMorphismInverse}$                             | IsClosedMonoidalCategory                       |
| for IsList, 25   | for IsCapCategory, 59                          |
| InternalCoHomTensorProduct-                                      | IsCoclosedMonoidalCategory                     |
| ${\tt CompatibilityMorphismInverse-}$                            | for IsCapCategory, 59                          |
| ${	t With Given Objects}$  | IsMonoidalCategory                             |
| for IsCapCategoryObject, IsList, IsCapCate-                      | for IsCapCategory, 59                          |
| goryObject, 26   | IsomorphismFromCoDualObjectToInternal-         |
| InternalCoHomTensorProduct-                                      | CoHomFromTensorUnit                            |
| CompatibilityMorphismWith-                                       | for IsCapCategoryObject, 20                    |
| GivenObjects   | IsomorphismFromDualObjectToInternal-           |
| for IsCapCategoryObject, IsList, IsCapCate-                      | HomIntoTensorUnit                              |
| goryObject, 19   | for IsCapCategoryObject, 14                    |
| InternalCoHomToTensorProduct-                                    | IsomorphismFromInternalCoHomFrom-              |
| ${\tt AdjunctionMap}$  | TensorUnitToCoDualObject                       |
| for IsCapCategoryObject, IsCapCategory-                          | for IsCapCategoryObject, 20                    |
| Object, IsCapCategoryMorphism, 17                                | IsomorphismFromInternalCoHomToObject           |
| InternalCoHomToTensorProduct-                                    | for IsCapCategoryObject, 21                    |
| ${\tt AdjunctionMapWithGivenTensor-}$                            | IsomorphismFromInternalCoHomToObject-          |
| Product  | WithGivenInternalCoHom                         |
| for IsCapCategoryObject, IsCapCategory-                          | for IsCapCategoryObject, IsCapCategory-        |
| Object, IsCapCategoryMorphism, Is-                               | Object, 21                                     |
| CapCategoryObject, 17  | IsomorphismFromInternalCoHomToTensor-          |
| InternalHom  | ProductWithCoDualObject                        |
| for IsCapCategoryCell, IsCapCategoryCell,                        | for IsCapCategoryObject, IsCapCategory-        |
| 27   | Object, 25                                     |
| InternalHomOnMorphisms   | Ουject, 25                                     |

| IsomorphismFromInternalHomIntoTensor-                 | LambdaElimination                                |
|---|--|
| ${\tt UnitToDualObject}$                              | for IsCapCategoryObject, IsCapCategory           |
| for IsCapCategoryObject, 14                           | Object, IsCapCategoryMorphism, 14                |
| IsomorphismFromInternalHomToObject                    | LambdaIntroduction                               |
| for IsCapCategoryObject, 15                           | for IsCapCategoryMorphism, 14                    |
| IsomorphismFromInternalHomToObject-                   | LeftDistributivityExpanding                      |
| WithGivenInternalHom                                  | for IsCapCategoryObject, IsList, 6               |
| for IsCapCategoryObject, IsCapCategory-               | LeftDistributivityExpandingWithGiven-            |
| Object, 15  | Objects  |
| IsomorphismFromInternalHomToTensor-                   | for IsCapCategoryObject, IsCapCategory           |
| ProductWithDualObject                                 | Object, IsList, IsCapCategoryObject, 6           |
| for IsCapCategoryObject, IsCapCategory-               | LeftDistributivityFactoring                      |
| Object, 22  | for IsCapCategoryObject, IsList, 6               |
| IsomorphismFromObjectToInternalCoHom                  | LeftDistributivityFactoringWithGiven-            |
| for IsCapCategoryObject, 21                           |  |
|   | Objects  for IsConCotogoryObject IsConCotogory   |
| IsomorphismFromObjectToInternalCoHom-                 | for IsCapCategoryObject, IsCapCategory           |
| WithGivenInternalCoHom                                | Object, IsList, IsCapCategoryObject, 7           |
| for IsCapCategoryObject, IsCapCategory-               | LeftUnitor                                       |
| Object, 21  | for IsCapCategoryObject, 4                       |
| IsomorphismFromObjectToInternalHom                    | LeftUnitorInverse                                |
| for IsCapCategoryObject, 14                           | for IsCapCategoryObject, 5                       |
| ${\tt IsomorphismFromObjectToInternalHom-}$           | LeftUnitorInverseWithGivenTensor-                |
| ${\tt WithGivenInternalHom}$                          | Product  |
| for IsCapCategoryObject, IsCapCategory-<br>Object, 15 | for IsCapCategoryObject, IsCapCategory-Object, 5 |
| IsomorphismFromTensorProductWithCo-                   | LeftUnitorWithGivenTensorProduct                 |
| DualObjectToInternalCoHom                             | for IsCapCategoryObject, IsCapCategory           |
| for IsCapCategoryObject, IsCapCategory-               | Object, 5  |
| Object, 25  | <b>3</b>   |
| IsomorphismFromTensorProductWithDual-                 | ${\tt MonoidalCategoriesTensorProductAnd-}$      |
| ObjectToInternalHom                                   | UnitTest, 53                                     |
| for IsCapCategoryObject, IsCapCategory-               | MonoidalCategoriesTest, 53                       |
| Object, 22  | ${\tt MonoidalPostCoComposeMorphism}$            |
| IsRigidSymmetricClosedMonoidalCategory                | for IsCapCategoryObject, IsCapCategory           |
| for IsCapCategory, 60                                 | Object, IsCapCategoryObject, 18                  |
| IsRigidSymmetricCoclosedMonoidal-                     | MonoidalPostCoComposeMorphismWith-               |
| Category  | GivenObjects                                     |
| for IsCapCategory, 60                                 | for IsCapCategoryObject, IsCapCategory-          |
| IsStrictMonoidalCategory                              | Object, IsCapCategoryObject, IsCap               |
| _ · ·   | CategoryObject, IsCapCategoryObject              |
| for IsCapCategory, 59                                 | 18   |
| IsSymmetricClosedMonoidalCategory                     | MonoidalPostComposeMorphism                      |
| for IsCapCategory, 60                                 | for IsCapCategoryObject, IsCapCategory           |
| IsSymmetricCoclosedMonoidalCategory                   | Object, IsCapCategoryObject, 11                  |
| for IsCapCategory, 60                                 | MonoidalPostComposeMorphismWithGiven-            |
| IsSymmetricMonoidalCategory                           | Objects  |
| for IsCapCategory, 60                                 | onlects  |

for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, IsCap-Object, IsCapCategoryObject, IsCap-CategoryObject, IsCapCategoryObject, CategoryObject, 23 MorphismFromTensorProductToInternal-11  ${\tt MonoidalPreCoComposeMorphism}$ CoHom for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 17 Object, 25 MonoidalPreCoComposeMorphismWithGiven-MorphismFromTensorProductToInternalCo-Objects HomWithGivenObjects for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, IsCap-Object, IsCapCategoryObject, IsCap-CategoryObject, IsCapCategoryObject, CategoryObject, 25 17  ${\tt MorphismFromTensorProductToInternalHom}$ MonoidalPreComposeMorphism for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, 13 Object, IsCapCategoryObject, 11 MorphismFromTensorProductToInternal- ${\tt Monoidal PreComposeMorphismWithGiven-}$ HomWithGivenObjects Objects for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, IsCap-Object, IsCapCategoryObject, IsCap-CategoryObject, 13 CategoryObject, IsCapCategoryObject, MorphismToBidual 11 for IsCapCategoryObject, 12  ${\tt MorphismFromBidual}$ MorphismToBidualWithGivenBidual for IsCapCategoryObject, 24 for IsCapCategoryObject, IsCapCategory- ${\tt MorphismFromBidualWithGivenBidual}$ Object, 12 for IsCapCategoryObject, IsCapCategory-MorphismToCoBidual Object, 24 for IsCapCategoryObject, 26 MorphismFromCoBidual MorphismToCoBidualWithGivenCoBidual for IsCapCategoryObject, 19 for IsCapCategoryObject, IsCapCategory-MorphismFromCoBidualWithGivenCoBidual Object, 27 for IsCapCategoryObject, IsCapCategory-RankMorphism Object, 19 for IsCapCategoryObject, 24  ${\tt MorphismFromInternalCoHomToTensor-}$ RightDistributivityExpanding Product for IsList, IsCapCategoryObject, 7 for IsCapCategoryObject, IsCapCategory-RightDistributivityExpandingWithGiven-Object, 20 Objects MorphismFromInternalCoHomToTensorfor IsCapCategoryObject, IsList, IsCapCate-ProductWithGivenObjects goryObject, IsCapCategoryObject, 7 for IsCapCategoryObject, IsCapCategory-RightDistributivityFactoring Object, IsCapCategoryObject, IsCapfor IsList, IsCapCategoryObject, 7 CategoryObject, 20 RightDistributivityFactoringWithGiven- ${\tt MorphismFromInternalHomToTensorProduct}$ Objects for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsList, IsCapCate-Object, 23 goryObject, IsCapCategoryObject, 7 MorphismFromInternalHomToTensor-RightUnitor ProductWithGivenObjects for IsCapCategoryObject, 5

| RightUnitorInverse                                    | for IsCapCategoryObject, IsCapCategory- |
|---|---|
| for IsCapCategoryObject, 5                            | Object, 6                               |
| RightUnitorInverseWithGivenTensor-                    | ${\tt TensorProductToInternalCoHom-}$   |
| Product   | ${\tt AdjunctionMap}$                   |
| for IsCapCategoryObject, IsCapCategory-               | for IsCapCategoryObject, IsCapCategory- |
| Object, 5   | Object, IsCapCategoryMorphism, 16       |
| ${\tt RightUnitorWithGivenTensorProduct}$             | TensorProductToInternalCoHom-           |
| for IsCapCategoryObject, IsCapCategory-               | ${\tt AdjunctionMapWithGivenInternal-}$ |
| Object, 5   | CoHom                                   |
| RigidSymmetricClosedMonoidal-                         | for IsCapCategoryObject, IsCapCategory- |
| CategoriesTest, 54                                    | Object, IsCapCategoryMorphism, Is-      |
| RigidSymmetricCoclosedMonoidal-                       | CapCategoryObject, 17                   |
| CategoriesTest, 54                                    | TensorProductToInternalHom-             |
| Ton combine de et Decelit es General de la litera     | ${\tt AdjunctionMap}$                   |
| TensorProductDualityCompatibility-                    | for IsCapCategoryObject, IsCapCategory- |
| Morphism  for IsConCotogory/Object IsConCotogory      | Object, IsCapCategoryMorphism, 10       |
| for IsCapCategoryObject, IsCapCategory-<br>Object, 13 | TensorProductToInternalHomAdjunction-   |
| TensorProductDualityCompatibility-                    | MapWithGivenInternalHom                 |
| MorphismWithGivenObjects                              | for IsCapCategoryObject, IsCapCategory- |
| for IsCapCategoryObject, IsCapCategory-               | Object, IsCapCategoryMorphism, Is-      |
| Object, IsCapCategoryObject, IsCap-                   | CapCategoryObject, 10                   |
| CategoryObject, 13                                    | TensorUnit                              |
| TensorProductInternalHomCompatibility-                | for IsCapCategory, 6                    |
| Morphism  | TraceMap                                |
| for IsList, 12  | for IsCapCategoryMorphism, 24           |
| TensorProductInternalHomCompatibility-                | UniversalPropertyOfCoDual               |
| MorphismInverse                                       | for IsCapCategoryObject, IsCapCategory- |
| for IsList, 23  | Object, IsCapCategoryMorphism, 21       |
| TensorProductInternalHomCompatibility-                | UniversalPropertyOfDual                 |
| MorphismInverseWithGivenObjects                       | for IsCapCategoryObject, IsCapCategory- |
| for IsCapCategoryObject, IsList, IsCapCate-           | Object, IsCapCategoryMorphism, 14       |
| goryObject, 23  |   |
| TensorProductInternalHomCompatibility-                | WriteFileForClosedMonoidalStructure, 57 |
| MorphismWithGivenObjects                              | WriteFileForCoclosedMonoidalStructure,  |
| for IsCapCategoryObject, IsList, IsCapCate-           | 57                                      |
| goryObject, 13  | WriteFileForMonoidalStructure, 57       |
| TensorProductOnMorphisms                              |   |
| for IsCapCategoryMorphism, IsCapCatego-               |   |
| ryMorphism, 3   |   |
| TensorProductOnMorphismsWithGiven-                    |   |
| TensorProducts  |   |
| for IsCapCategoryObject, IsCapCategory-               |   |
| Morphism, IsCapCategoryMorphism,                      |   |
| IsCapCategoryObject, 4                                |   |
| TensorProductOnObjects                                |   |