MonoidalCategories

Monoidal and monoidal (co)closed categories

2019.06.07

7 June 2019

Mohamed Barakat

Sebastian Gutsche

Sebastian Posur

Mohamed Barakat
Email: mohamed.barakat@uni-siegen.de
Homepage: http://algebra.mathematik.uni-siegen.de/barakat/
Address: Walter-Flex-Str. 3
57068 Siegen
Germany

Sebastian Gutsche
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

Contents

Chapter 1

Tensor Product and Internal Hom

1.1 Monoidal Categories

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

- a category $\mathbf{C}$,
- a functor $\otimes : \mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$,
- an object $1 \in \mathbf{C}$,
- 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: $(\alpha_{a,b,c} \otimes \mathrm{id}_d) \circ \alpha_{a,b \otimes c,d} \circ (\mathrm{id}_a \otimes \alpha_{b,c,d}) = \alpha_{a \otimes b,c,d} \circ \alpha_{a,b,c \otimes d}$,
- for all objects a,c, the triangle identity holds: $(\rho_a \otimes \mathrm{id}_c) \circ \alpha_{a,1,c} = \mathrm{id}_a \otimes \lambda_c$.

The corresponding GAP property is given by `IsMonoidalCategory`.

1.1.1 TensorProductOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ `TensorProductOnObjects(a, b)` (operation)

Returns: an object

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

1.1.2 AddTensorProductOnObjects (for IsCapCategory, IsFunction)

▷ `AddTensorProductOnObjects(C, F)` (operation)

Returns: nothing

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

1.1.3 TensorProductOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ `TensorProductOnMorphisms(alpha, beta)` (operation)

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

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

1.1.4 TensorProductOnMorphismsWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

▷ `TensorProductOnMorphismsWithGivenTensorProducts(s, alpha, beta, r)` (operation)

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

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 $\alpha \otimes \beta$.

1.1.5 AddTensorProductOnMorphismsWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ `AddTensorProductOnMorphismsWithGivenTensorProducts(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TensorProductOnMorphismsWithGivenTensorProducts`. $F : (a \otimes b, \alpha : a \to a', \beta : b \to b', a' \otimes b') \mapsto \alpha \otimes \beta$.

1.1.6 AssociatorRightToLeft (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `AssociatorRightToLeft(a, b, c)` (operation)

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

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

1.1.7 AssociatorRightToLeftWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `AssociatorRightToLeftWithGivenTensorProducts(s, a, b, c, r)` (operation)

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

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 $\alpha_{a,(b,c)} : a \otimes (b \otimes c) \to (a \otimes b) \otimes c$.

1.1.8 AddAssociatorRightToLeftWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ `AddAssociatorRightToLeftWithGivenTensorProducts(C, F)` (operation)

Returns: nothing

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

1.1.9 AssociatorLeftToRight (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `AssociatorLeftToRight(a, b, c)` (operation)

Returns: a morphism in $\mathrm{Hom}((a \otimes b) \otimes c \to a \otimes (b \otimes c))$.

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

1.1.10 AssociatorLeftToRightWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `AssociatorLeftToRightWithGivenTensorProducts(s, a, b, c, r)` (operation)

Returns: a morphism in $\mathrm{Hom}((a \otimes b) \otimes c \to a \otimes (b \otimes c))$.

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 $\alpha_{(a,b),c} : (a \otimes b) \otimes c \to a \otimes (b \otimes c)$.

1.1.11 AddAssociatorLeftToRightWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ `AddAssociatorLeftToRightWithGivenTensorProducts(C, F)` (operation)

Returns: nothing

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

1.1.12 TensorUnit (for IsCapCategory)

▷ `TensorUnit(C)` (attribute)

Returns: an object

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

1.1.13 AddTensorUnit (for IsCapCategory, IsFunction)

▷ `AddTensorUnit(C, F)` (operation)

Returns: nothing

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

1.1.14 LeftUnitor (for IsCapCategoryObject)

▷ `LeftUnitor(a)` (attribute)

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

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

1.1.15 LeftUnitorWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `LeftUnitorWithGivenTensorProduct(a, s)` (operation)

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

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

1.1.16 AddLeftUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddLeftUnitorWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LeftUnitorWithGivenTensorProduct`. $F : (a, 1 \otimes a) \mapsto \lambda_a$.

1.1.17 LeftUnitorInverse (for IsCapCategoryObject)

▷ `LeftUnitorInverse(a)` (attribute)

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

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

1.1.18 LeftUnitorInverseWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `LeftUnitorInverseWithGivenTensorProduct(a, r)` (operation)

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

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

1.1.19 AddLeftUnitorInverseWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddLeftUnitorInverseWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LeftUnitorInverseWithGivenTensorProduct`. $F : (a, 1 \otimes a) \mapsto \lambda_a^{-1}$.

1.1.20 RightUnitor (for IsCapCategoryObject)

▷ `RightUnitor(a)` (attribute)

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

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

1.1.21 RightUnitorWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `RightUnitorWithGivenTensorProduct(a, s)` (operation)

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

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

1.1.22 AddRightUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddRightUnitorWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `RightUnitorWithGivenTensorProduct`. $F : (a, a \otimes 1) \mapsto \rho_a$.

1.1.23 RightUnitorInverse (for IsCapCategoryObject)

▷ `RightUnitorInverse(a)` (attribute)

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

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

1.1.24 RightUnitorInverseWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `RightUnitorInverseWithGivenTensorProduct(a, r)` (operation)

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

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

1.1.25 AddRightUnitorInverseWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddRightUnitorInverseWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `RightUnitorInverseWithGivenTensorProduct`. $F : (a, a \otimes 1) \mapsto \rho_a^{-1}$.

1.1.26 LeftDistributivityExpanding (for IsCapCategoryObject, IsList)

▷ `LeftDistributivityExpanding(a, L)` (operation)

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

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

1.1.27 LeftDistributivityExpandingWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsList, IsCapCategoryObject)

▷ `LeftDistributivityExpandingWithGivenObjects(s, a, L, r)` (operation)

Returns: a morphism in $\mathrm{Hom}(s, r)$

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

1.1.28 AddLeftDistributivityExpandingWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddLeftDistributivityExpandingWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LeftDistributivityExpandingWithGivenObjects`. $F : (a \otimes (b_1 \oplus \cdots \oplus b_n), a, L, (a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n)) \mapsto \mathrm{LeftDistributivityExpandingWithGivenObjects}(a, L)$.

1.1.29 LeftDistributivityFactoring (for IsCapCategoryObject, IsList)

▷ `LeftDistributivityFactoring(a, L)` (operation)

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

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

1.1.30 LeftDistributivityFactoringWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsList, IsCapCategoryObject)

▷ `LeftDistributivityFactoringWithGivenObjects(s, a, L, r)` (operation)

Returns: a morphism in $\mathrm{Hom}(s, r)$

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

1.1.31 AddLeftDistributivityFactoringWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddLeftDistributivityFactoringWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LeftDistributivityFactoringWithGivenObjects`. $F : ((a \otimes b_1) \oplus \cdots \oplus (a \otimes b_n), a, L, a \otimes (b_1 \oplus \cdots \oplus b_n)) \mapsto \mathrm{LeftDistributivityFactoringWithGivenObjects}(a, L)$.

1.1.32 RightDistributivityExpanding (for IsList, IsCapCategoryObject)

▷ `RightDistributivityExpanding(L, a)` (operation)

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

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

1.1.33 RightDistributivityExpandingWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject, IsCapCategoryObject)

▷ `RightDistributivityExpandingWithGivenObjects(s, L, a, r)` (operation)

Returns: a morphism in $\mathrm{Hom}(s, r)$

The arguments are an object $s = (b_1 \oplus \cdots \oplus b_n) \otimes a$, a list of objects $L = (b_1, \dots, b_n)$, 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 $s \to r$.

1.1.34 AddRightDistributivityExpandingWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddRightDistributivityExpandingWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `RightDistributivityExpandingWithGivenObjects`. $F : ((b_1 \oplus \dots \oplus b_n) \otimes a, L, a, (b_1 \otimes a) \oplus \dots \oplus (b_n \otimes a)) \mapsto \mathrm{RightDistributivityExpandingWithGivenObjects}(L, a)$.

1.1.35 RightDistributivityFactoring (for IsList, IsCapCategoryObject)

▷ `RightDistributivityFactoring(L, a)` (operation)

Returns: a morphism in $\mathrm{Hom}((b_1 \otimes a) \oplus \dots \oplus (b_n \otimes a), (b_1 \oplus \dots \oplus b_n) \otimes a)$

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

1.1.36 RightDistributivityFactoringWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject, IsCapCategoryObject)

▷ `RightDistributivityFactoringWithGivenObjects(s, L, a, r)` (operation)

Returns: a morphism in $\mathrm{Hom}(s, r)$

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

1.1.37 AddRightDistributivityFactoringWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddRightDistributivityFactoringWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `RightDistributivityFactoringWithGivenObjects`. $F : ((b_1 \otimes a) \oplus \dots \oplus (b_n \otimes a), L, a, (b_1 \oplus \dots \oplus b_n) \otimes a) \mapsto \mathrm{RightDistributivityFactoringWithGivenObjects}(L, a)$.

1.2 Braided Monoidal Categories

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

- $\lambda_a \circ B_{a,1} = \rho_a$,
- $(B_{c,a} \otimes \mathrm{id}_b) \circ \alpha_{c,a,b} \circ B_{a \otimes b,c} = \alpha_{a,c,b} \circ (\mathrm{id}_a \otimes B_{b,c}) \circ \alpha^{-1}_{a,b,c}$,
- $(\mathrm{id}_b \otimes B_{c,a}) \circ \alpha^{-1}_{b,c,a} \circ B_{a,b \otimes c} = \alpha^{-1}_{b,a,c} \circ (B_{a,b} \otimes \mathrm{id}_c) \circ \alpha_{a,b,c}$.

The corresponding GAP property is given by `IsBraidedMonoidalCategory`.

1.2.1 Braiding (for IsCapCategoryObject, IsCapCategoryObject)

▷ `Braiding(a, b)` (operation)

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

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

1.2.2 BraidingWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `BraidingWithGivenTensorProducts(s, a, b, r)` (operation)

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

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 $B_{a,b} : a \otimes b \to b \otimes a$.

1.2.3 AddBraidingWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ `AddBraidingWithGivenTensorProducts(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `BraidingWithGivenTensorProducts`. $F : (a \otimes b, a, b, b \otimes a) \to B_{a,b}$.

1.2.4 BraidingInverse (for IsCapCategoryObject, IsCapCategoryObject)

▷ `BraidingInverse(a, b)` (operation)

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

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

1.2.5 BraidingInverseWithGivenTensorProducts (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `BraidingInverseWithGivenTensorProducts(s, a, b, r)` (operation)

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

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

1.2.6 AddBraidingInverseWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ `AddBraidingInverseWithGivenTensorProducts(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `BraidingInverseWithGivenTensorProducts`. $F : (b \otimes a, a, b, a \otimes b) \to B_{a,b}^{-1}$.

1.3 Symmetric Monoidal Categories

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

1.4 Symmetric Closed Monoidal Categories

A symmetric monoidal category **C** which has for each functor $-\otimes b : \mathbf{C} \rightarrow \mathbf{C}$ a right adjoint (denoted by $\underline{\mathrm{Hom}}(b,-)$) is called a *symmetric closed monoidal category*. The corresponding GAP property is given by `IsSymmetricClosedMonoidalCategory`.

1.4.1 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 $\underline{\mathrm{Hom}}(a,b)$.

1.4.2 AddInternalHomOnObjects (for IsCapCategory, IsFunction)

▷ `AddInternalHomOnObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `InternalHomOnObjects`. $F : (a,b) \mapsto \underline{\mathrm{Hom}}(a,b)$.

1.4.3 InternalHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ `InternalHomOnMorphisms(alpha, beta)` (operation)

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

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

1.4.4 InternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

▷ `InternalHomOnMorphismsWithGivenInternalHoms(s, alpha, beta, r)` (operation)

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

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

1.4.5 AddInternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategory, IsFunction)

▷ `AddInternalHomOnMorphismsWithGivenInternalHoms(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `InternalHomOnMorphismsWithGivenInternalHoms`. $F : (\underline{\mathrm{Hom}}(a',b), \alpha : a \rightarrow a', \beta : b \rightarrow b', \underline{\mathrm{Hom}}(a,b')) \mapsto \underline{\mathrm{Hom}}(\alpha,\beta)$.

1.4.6 EvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ `EvaluationMorphism(a, b)` (operation)

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

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

1.4.7 EvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `EvaluationMorphismWithGivenSource(a, b, s)` (operation)

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

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

1.4.8 AddEvaluationMorphismWithGivenSource (for IsCapCategory, IsFunction)

▷ `AddEvaluationMorphismWithGivenSource(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `EvaluationMorphismWithGivenSource`. $F : (a,b,\underline{\mathrm{Hom}}(a,b)\otimes a) \mapsto \mathrm{ev}_{a,b}$.

1.4.9 CoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ `CoevaluationMorphism(a, b)` (operation)

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

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

1.4.10 CoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `CoevaluationMorphismWithGivenRange(a, b, r)` (operation)

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

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

1.4.11 AddCoevaluationMorphismWithGivenRange (for IsCapCategory, IsFunction)

▷ `AddCoevaluationMorphismWithGivenRange(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `CoevaluationMorphismWithGivenRange`. $F : (a,b,\underline{\mathrm{Hom}}(b,a\otimes b)) \mapsto \mathrm{coev}_{a,b}$.

1.4.12 TensorProductToInternalHomAdjunctionMap (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ `TensorProductToInternalHomAdjunctionMap(a, b, f)` (operation)

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

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

1.4.13 AddTensorProductToInternalHomAdjunctionMap (for IsCapCategory, IsFunction)

▷ `AddTensorProductToInternalHomAdjunctionMap(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TensorProductToInternalHomAdjunctionMap`. $F : (a,b,f : a \otimes b \to c) \mapsto (g : a \to \underline{\mathrm{Hom}}(b,c))$.

1.4.14 InternalHomToTensorProductAdjunctionMap (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ `InternalHomToTensorProductAdjunctionMap(b, c, g)` (operation)

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

The arguments are 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.

1.4.15 AddInternalHomToTensorProductAdjunctionMap (for IsCapCategory, IsFunction)

▷ `AddInternalHomToTensorProductAdjunctionMap(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `InternalHomToTensorProductAdjunctionMap`. $F : (b,c,g : a \to \underline{\mathrm{Hom}}(b,c)) \mapsto (g : a \otimes b \to c)$.

1.4.16 MonoidalPreComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MonoidalPreComposeMorphism(a, b, c)` (operation)

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

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

1.4.17 MonoidalPreComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MonoidalPreComposeMorphismWithGivenObjects(s, a, b, c, r)` (operation)

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

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 $\mathrm{MonoidalPreComposeMorphismWithGivenObjects}_{a,b,c} : \underline{\mathrm{Hom}}(a,b) \otimes \underline{\mathrm{Hom}}(b,c) \to \underline{\mathrm{Hom}}(a,c)$.

1.4.18 AddMonoidalPreComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddMonoidalPreComposeMorphismWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `MonoidalPreComposeMorphismWithGivenObjects`. $F : (\underline{\mathrm{Hom}}(a,b) \otimes \underline{\mathrm{Hom}}(b,c), a, b, c, \underline{\mathrm{Hom}}(a,c)) \mapsto \mathrm{MonoidalPreComposeMorphismWithGivenObjects}_{a,b,c}$.

1.4.19 MonoidalPostComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MonoidalPostComposeMorphism(a, b, c)` (operation)

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

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

1.4.20 MonoidalPostComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MonoidalPostComposeMorphismWithGivenObjects(s, a, b, c, r)` (operation)

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

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 $\mathrm{MonoidalPostComposeMorphismWithGivenObjects}_{a,b,c} : \underline{\mathrm{Hom}}(b,c) \otimes \underline{\mathrm{Hom}}(a,b) \to \underline{\mathrm{Hom}}(a,c)$.

1.4.21 AddMonoidalPostComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddMonoidalPostComposeMorphismWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `MonoidalPostComposeMorphismWithGivenObjects`. $F : (\underline{\mathrm{Hom}}(b,c) \otimes \underline{\mathrm{Hom}}(a,b), a, b, c, \underline{\mathrm{Hom}}(a,c)) \mapsto \mathrm{MonoidalPostComposeMorphismWithGivenObjects}_{a,b,c}$.

1.4.22 DualOnObjects (for IsCapCategoryObject)

▷ `DualOnObjects(a)` (attribute)

Returns: an object

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

1.4.23 AddDualOnObjects (for IsCapCategory, IsFunction)

▷ `AddDualOnObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `DualOnObjects`. $F : a \mapsto a^{\vee}$.

1.4.24 DualOnMorphisms (for IsCapCategoryMorphism)

▷ `DualOnMorphisms(alpha)` (attribute)

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

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

1.4.25 DualOnMorphismsWithGivenDuals (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

▷ `DualOnMorphismsWithGivenDuals(s, alpha, r)` (operation)

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

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

1.4.26 AddDualOnMorphismsWithGivenDuals (for IsCapCategory, IsFunction)

▷ `AddDualOnMorphismsWithGivenDuals(C, F)` (operation)

Returns: nothing

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

1.4.27 EvaluationForDual (for IsCapCategoryObject)

▷ `EvaluationForDual(a)` (attribute)

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

The argument is an object a. The output is the evaluation morphism $\mathrm{ev}_a : a^{\vee} \otimes a \rightarrow 1$.

1.4.28 EvaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `EvaluationForDualWithGivenTensorProduct(s, a, r)` (operation)

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

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

1.4.29 AddEvaluationForDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddEvaluationForDualWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `EvaluationForDualWithGivenTensorProduct`. $F : (a^\vee \otimes a, a, 1) \mapsto \mathrm{ev}_a$.

1.4.30 CoevaluationForDual (for IsCapCategoryObject)

▷ `CoevaluationForDual(a)` (attribute)

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

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

1.4.31 CoevaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `CoevaluationForDualWithGivenTensorProduct(s, a, r)` (operation)

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

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

1.4.32 AddCoevaluationForDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

▷ `AddCoevaluationForDualWithGivenTensorProduct(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `CoevaluationForDualWithGivenTensorProduct`. $F : (1, a, a \otimes a^\vee) \mapsto \mathrm{coev}_a$.

1.4.33 MorphismToBidual (for IsCapCategoryObject)

▷ `MorphismToBidual(a)` (attribute)

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

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

1.4.34 MorphismToBidualWithGivenBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismToBidualWithGivenBidual(a, r)` (operation)

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

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

1.4.35 AddMorphismToBidualWithGivenBidual (for IsCapCategory, IsFunction)

▷ `AddMorphismToBidualWithGivenBidual(C, F)` (operation)

Returns: nothing

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

1.4.36 TensorProductInternalHomCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `TensorProductInternalHomCompatibilityMorphism(a, a', b, b')` (operation)

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

The arguments are four objects a, a', b, b'. The output is the natural morphism $\mathrm{TensorProductInternalHomCompatibilityMorphismWithGivenObjects}_{a,a',b,b'} : \underline{\mathrm{Hom}}(a,a') \otimes \underline{\mathrm{Hom}}(b,b') \rightarrow \underline{\mathrm{Hom}}(a \otimes b, a' \otimes b')$.

1.4.37 TensorProductInternalHomCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsList)

▷ `TensorProductInternalHomCompatibilityMorphismWithGivenObjects(a, a', b, b', L)` (operation)

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

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

1.4.38 AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TensorProductInternalHomCompatibilityMorphismWithGivenObjects`. $F : (a, a', b, b', [\underline{\mathrm{Hom}}(a,a') \otimes \underline{\mathrm{Hom}}(b,b'), \underline{\mathrm{Hom}}(a \otimes b, a' \otimes b')]) \mapsto \mathrm{TensorProductInternalHomCompatibilityMorphismWithGivenObjects}_{a,a',b,b'}$.

1.4.39 TensorProductDualityCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ `TensorProductDualityCompatibilityMorphism(a, b)` (operation)

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

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

1.4.40 TensorProductDualityCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `TensorProductDualityCompatibilityMorphismWithGivenObjects(s, a, b, r)` (operation)

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

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$_{a,b}: a^\vee \otimes b^\vee \rightarrow (a \otimes b)^\vee$.

1.4.41 AddTensorProductDualityCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddTensorProductDualityCompatibilityMorphismWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TensorProductDualityCompatibilityMorphismWithGivenObjects`. $F : (a^\vee \otimes b^\vee, a, b, (a \otimes b)^\vee) \mapsto$ TensorProductDualityCompatibilityMorphismWithGivenObjects$_{a,b}$.

1.4.42 MorphismFromTensorProductToInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismFromTensorProductToInternalHom(a, b)` (operation)

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

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

1.4.43 MorphismFromTensorProductToInternalHomWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismFromTensorProductToInternalHomWithGivenObjects(s, a, b, r)` (operation)

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

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$_{a,b}: a^\vee \otimes b \rightarrow \underline{\mathrm{Hom}}(a,b)$.

1.4.44 AddMorphismFromTensorProductToInternalHomWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddMorphismFromTensorProductToInternalHomWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `MorphismFromTensorProductToInternalHomWithGivenObjects`. $F : (a^\vee \otimes b, a, b, \underline{\mathrm{Hom}}(a,b)) \mapsto$ MorphismFromTensorProductToInternalHomWithGivenObjects$_{a,b}$.

1.4.45 IsomorphismFromTensorProductToInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ `IsomorphismFromTensorProductToInternalHom(a, b)` (operation)

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

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

1.4.46 AddIsomorphismFromTensorProductToInternalHom (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromTensorProductToInternalHom(C, F)` (operation)

Returns: nothing

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

1.4.47 MorphismFromInternalHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismFromInternalHomToTensorProduct(a, b)` (operation)

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

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

1.4.48 MorphismFromInternalHomToTensorProductWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismFromInternalHomToTensorProductWithGivenObjects(s, a, b, r)` (operation)

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

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 $\mathrm{MorphismFromInternalHomToTensorProductWithGivenObjects}_{a,b} : \underline{\mathrm{Hom}}(a,b) \rightarrow a^\vee \otimes b$.

1.4.49 AddMorphismFromInternalHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddMorphismFromInternalHomToTensorProductWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `MorphismFromInternalHomToTensorProductWithGivenObjects`. $F : (\underline{\mathrm{Hom}}(a,b), a, b, a^\vee \otimes b) \mapsto \mathrm{MorphismFromInternalHomToTensorProductWithGivenObjects}_{a,b}$.

1.4.50 IsomorphismFromInternalHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ `IsomorphismFromInternalHomToTensorProduct(a, b)` (operation)

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

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

1.4.51 AddIsomorphismFromInternalHomToTensorProduct (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromInternalHomToTensorProduct(C, F)` (operation)

Returns: nothing

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

1.4.52 TraceMap (for IsCapCategoryMorphism)

▷ `TraceMap(alpha)` (attribute)

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

The argument is a morphism α. The output is the trace morphism $\text{trace}_\alpha : 1 \to 1$.

1.4.53 AddTraceMap (for IsCapCategory, IsFunction)

▷ `AddTraceMap(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TraceMap`. $F : \alpha \mapsto \text{trace}_\alpha$

1.4.54 RankMorphism (for IsCapCategoryObject)

▷ `RankMorphism(a)` (attribute)

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

The argument is an object a. The output is the rank morphism $\text{rank}_a : 1 \to 1$.

1.4.55 AddRankMorphism (for IsCapCategory, IsFunction)

▷ `AddRankMorphism(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `RankMorphism`. $F : a \mapsto \text{rank}_a$

1.4.56 IsomorphismFromDualToInternalHom (for IsCapCategoryObject)

▷ `IsomorphismFromDualToInternalHom(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 $\text{IsomorphismFromDualToInternalHom}_a : a^\vee \to \underline{\text{Hom}}(a,1)$.

1.4.57 AddIsomorphismFromDualToInternalHom (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromDualToInternalHom(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `IsomorphismFromDualToInternalHom`. $F : a \mapsto \mathrm{IsomorphismFromDualToInternalHom}_a$

1.4.58 IsomorphismFromInternalHomToDual (for IsCapCategoryObject)

▷ `IsomorphismFromInternalHomToDual(a)` (attribute)

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

The argument is an object a. The output is the isomorphism $\mathrm{IsomorphismFromInternalHomToDual}_a : \underline{\mathrm{Hom}}(a,1) \rightarrow a^{\vee}$.

1.4.59 AddIsomorphismFromInternalHomToDual (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromInternalHomToDual(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `IsomorphismFromInternalHomToDual`. $F : a \mapsto \mathrm{IsomorphismFromInternalHomToDual}_a$

1.4.60 UniversalPropertyOfDual (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ `UniversalPropertyOfDual(t, a, alpha)` (operation)

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

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

1.4.61 AddUniversalPropertyOfDual (for IsCapCategory, IsFunction)

▷ `AddUniversalPropertyOfDual(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `UniversalPropertyOfDual`. $F : (t,a,\alpha : t \otimes a \rightarrow 1) \mapsto (t \rightarrow a^{\vee})$.

1.4.62 LambdaIntroduction (for IsCapCategoryMorphism)

▷ `LambdaIntroduction(alpha)` (attribute)

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

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

1.4.63 AddLambdaIntroduction (for IsCapCategory, IsFunction)

▷ `AddLambdaIntroduction(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LambdaIntroduction`. $F : (\alpha : a \to b) \mapsto (1 \to \underline{\mathrm{Hom}}(a,b))$.

1.4.64 LambdaElimination (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ `LambdaElimination(a, b, alpha)` (operation)

Returns: a morphism in $\mathrm{Hom}(a,b)$.

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

1.4.65 AddLambdaElimination (for IsCapCategory, IsFunction)

▷ `AddLambdaElimination(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `LambdaElimination`. $F : (a,b,\alpha : 1 \to \underline{\mathrm{Hom}}(a,b)) \mapsto (a \to b)$.

1.4.66 IsomorphismFromObjectToInternalHom (for IsCapCategoryObject)

▷ `IsomorphismFromObjectToInternalHom(a)` (attribute)

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

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

1.4.67 IsomorphismFromObjectToInternalHomWithGivenInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ `IsomorphismFromObjectToInternalHomWithGivenInternalHom(a, r)` (operation)

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

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

1.4.68 AddIsomorphismFromObjectToInternalHomWithGivenInternalHom (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromObjectToInternalHomWithGivenInternalHom(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `IsomorphismFromObjectToInternalHomWithGivenInternalHom`. $F : (a, \underline{\mathrm{Hom}}(1,a)) \mapsto (a \to \underline{\mathrm{Hom}}(1,a))$.

1.4.69 IsomorphismFromInternalHomToObject (for IsCapCategoryObject)

▷ `IsomorphismFromInternalHomToObject(a)` (attribute)

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

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

1.4.70 IsomorphismFromInternalHomToObjectWithGivenInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ `IsomorphismFromInternalHomToObjectWithGivenInternalHom(a, s)` (operation)

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

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

1.4.71 AddIsomorphismFromInternalHomToObjectWithGivenInternalHom (for IsCapCategory, IsFunction)

▷ `AddIsomorphismFromInternalHomToObjectWithGivenInternalHom(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `IsomorphismFromInternalHomToObjectWithGivenInternalHom`. $F : (a, \underline{\mathrm{Hom}}(1,a)) \mapsto (\underline{\mathrm{Hom}}(1,a) \to a)$.

1.5 Rigid Symmetric Closed Monoidal Categories

A symmetric closed monoidal category **C** satisfying

- the natural morphism $\underline{\mathrm{Hom}}(a_1,b_1) \otimes \underline{\mathrm{Hom}}(a_2,b_2) \to \underline{\mathrm{Hom}}(a_1 \otimes a_2, b_1 \otimes b_2)$ is an isomorphism,
- the natural morphism $a \to \underline{\mathrm{Hom}}(\underline{\mathrm{Hom}}(a,1),1)$ is an isomorphism

is called a *rigid symmetric closed monoidal category*.

1.5.1 TensorProductInternalHomCompatibilityMorphismInverse (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ `TensorProductInternalHomCompatibilityMorphismInverse(a, a', b, b')` (operation)

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

The arguments are four objects a, a', b, b'. The output is the natural morphism $\mathrm{TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects}_{a,a',b,b'} : \underline{\mathrm{Hom}}(a \otimes b, a' \otimes b') \to \underline{\mathrm{Hom}}(a,a') \otimes \underline{\mathrm{Hom}}(b,b')$.

1.5.2 TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsList)

▷ `TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects(a, a', b, b', L)` (operation)

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

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

1.5.3 AddTensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategory, IsFunction)

▷ `AddTensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects`. $F : (a, a', b, b', [\underline{\mathrm{Hom}}(a,a') \otimes \underline{\mathrm{Hom}}(b,b'), \underline{\mathrm{Hom}}(a \otimes b, a' \otimes b')]) \mapsto \mathrm{TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects}_{a,a',b,b'}$.

1.5.4 MorphismFromBidual (for IsCapCategoryObject)

▷ `MorphismFromBidual(a)` (attribute)

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

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

1.5.5 MorphismFromBidualWithGivenBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ `MorphismFromBidualWithGivenBidual(a, s)` (operation)

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

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

1.5.6 AddMorphismFromBidualWithGivenBidual (for IsCapCategory, IsFunction)

▷ `AddMorphismFromBidualWithGivenBidual(C, F)` (operation)

Returns: nothing

The arguments are a category C and a function F. This operations adds the given function F to the category for the basic operation `MorphismFromBidualWithGivenBidual`. $F : (a, (a^{\vee})^{\vee}) \mapsto ((a^{\vee})^{\vee} \rightarrow a)$.

Chapter 2

Examples and Tests

2.1 Basics

Example

```
gap> vecspaces := CreateCapCategory( "VectorSpaces" );;
gap> ReadPackage( "MonoidalCategories", "examples/VectorSpacesMonoidalCategory.gi" );;
gap> z := ZeroObject( vecspaces );;
gap> a := QVectorSpace( 1 );;
gap> b := QVectorSpace( 2 );;
gap> c := QVectorSpace( 3 );;
gap> alpha := VectorSpaceMorphism( a, [ [ 1, 0 ] ], b );;
gap> beta := VectorSpaceMorphism( b, [ [ 1, 0, 0 ], [ 0, 1, 0 ] ], c );;
gap> gamma := VectorSpaceMorphism( c, [ [ 0, 1, 1 ], [ 1, 0, 1 ], [ 1, 1, 0 ] ], c );;
gap> IsCongruentForMorphisms( TensorProductOnMorphisms( alpha, beta ), TensorProductOnMorphisms(
false
gap> IsOne( AssociatorRightToLeft( a, b, c ) );
true
gap> IsCongruentForMorphisms( gamma, LambdaElimination( c, c, LambdaIntroduction( gamma ) ) );
true
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
```

Index