Monoidal and monoidal (co)closed categories

2024.09-05

12 September 2024

Mohamed Barakat

Sebastian Gutsche

Sebastian Posur

Tom Kuhmichel

Fabian Zickgraf

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

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

Fabian Zickgraf

Email: fabian.zickgraf@uni-siegen.de Homepage: https://github.com/zickgraf/

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

Contents

1	Monoidal 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	Left Closed Monoidal Categories	8
	1.6	Closed Monoidal Categories	15
	1.7	Left Coclosed Monoidal Categories	24
	1.8	Coclosed Monoidal Categories	31
	1.9	Symmetric Closed Monoidal Categories	39
	1.10	Symmetric Coclosed Monoidal Categories	40
	1.11	Rigid Symmetric Closed Monoidal Categories	40
	1.12	Rigid Symmetric Coclosed Monoidal Categories	42
	1.13	Convenience Methods	45
	1.14	Add-methods	45
2	Examples and Tests		105
	2.1	Test functions	105
3	Cod	Code Generation for Monodial Categories	
	3.1	Monoidal Categories	113
	3.2	Closed Monoidal Categories	113
	3.3	Coclosed Monoidal Categories	114
4	The terminal category with multiple objects		115
	4.1	Constructors	115
	4.2	GAP Categories	115
5	Legacy Operations and Synonyms		110
	5.1	Legacy operations	116
	5.2	Synonyms for legacy operations	117
6	MonoidalCategories automatic generated documentation		
	6.1	MonoidalCategories automatic generated documentation of properties	120 120
Index			122

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 TensorUnit (for IsCapCategory)

▷ TensorUnit(C)

(attribute)

Returns: an object

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

1.2 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)

 ${\scriptstyle \vartriangleright} \ \ \, \mathsf{LeftDistributivityExpandingWithGivenObjects(\textit{s},\textit{a},\textit{L},\textit{r})}$

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

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 $L = (b_1, \dots, b_n)$. The output is the left distributivity morphism $(a \otimes b_1) \oplus \dots \oplus (a \otimes b_n) \to a \otimes (b_1 \oplus \dots \oplus b_n)$.

1.2.4 LeftDistributivityFactoringWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsList, IsCapCategoryObject)

▷ 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)

 ${\tt \triangleright \; RightDistributivityExpandingWithGivenObjects(s, \; \textit{L, } \; \textit{a, } \; \textit{r})}$

(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)

ightharpoonup RightDistributivityFactoringWithGivenObjects(s, L, a, r)

(operation)

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 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)

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)

▷ 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 Left 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 $\operatorname{Hom}_{\ell}(b,-)$) is called a *left closed monoidal category*.

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

The corresponding GAP property is called IsLeftClosedMonoidalCategory.

1.5.1 LeftInternalHomOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftInternalHomOnObjects(a, b)

(operation)

Returns: an object

The arguments are two objects a, b. The output is the internal hom object $\underline{\text{Hom}}_{\ell}(a, b)$.

1.5.2 LeftInternalHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategory-Morphism)

▷ LeftInternalHomOnMorphisms(alpha, beta)

(operation)

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

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

1.5.3 LeftInternalHomOnMorphismsWithGivenLeftInternalHoms (for IsCapCatego-ryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

 \triangleright LeftInternalHomOnMorphismsWithGivenLeftInternalHoms(s, alpha, beta, r) (operation)

Returns: a morphism in Hom(s, r)

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

1.5.4 LeftClosedMonoidalEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftClosedMonoidalEvaluationMorphism(a, b)

(operation)

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

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

1.5.5 LeftClosedMonoidalEvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftClosedMonoidalEvaluationMorphismWithGivenSource(a, b, s) (operation) **Returns:** a morphism in Hom(s,b).

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

1.5.6 LeftClosedMonoidalCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftClosedMonoidalCoevaluationMorphism(a, b)

(operation)

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

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

1.5.7 LeftClosedMonoidalCoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

1.5.8 TensorProductToLeftInternalHomAdjunctMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

 \triangleright TensorProductToLeftInternalHomAdjunctMorphism(a, b, f) (operation) **Returns:** a morphism in Hom(a, Hom_{ℓ}(b, c)).

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

1.5.9 TensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 \triangleright TensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom(a, b, f, i) (operation)

Returns: a morphism in Hom(a, i).

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

1.5.10 LeftInternalHomToTensorProductAdjunctMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

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

1.5.11 LeftInternalHomToTensorProductAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject) IsCapCategoryObject)

Returns: a morphism in Hom(t,c).

The arguments are two objects b, c, a morphism $g : a \to \underline{\operatorname{Hom}}_{\ell}(b, c)$ 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 LeftClosedMonoidalPreComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

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

1.5.13 LeftClosedMonoidalPreComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftClosedMonoidalPreComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in Hom(s,r).

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

1.5.14 LeftClosedMonoidalPostComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ LeftClosedMonoidalPostComposeMorphism(a, b, c)

(operation)

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

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

1.5.15 LeftClosedMonoidalPostComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftClosedMonoidalPostComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in Hom(s, r).

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

1.5.16 LeftDualOnObjects (for IsCapCategoryObject)

▷ LeftDualOnObjects(a)

(attribute)

Returns: an object

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

1.5.17 LeftDualOnMorphisms (for IsCapCategoryMorphism)

 \triangleright LeftDualOnMorphisms(alpha)

(attribute)

(operation)

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 LeftDualOnMorphismsWithGivenLeftDuals (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

▷ LeftDualOnMorphismsWithGivenLeftDuals(s, alpha, r)

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

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 LeftClosedMonoidalEvaluationForLeftDual (for IsCapCategoryObject)

▷ LeftClosedMonoidalEvaluationForLeftDual(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 LeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct (for Is-CapCategoryObject, IsCapCategoryObject)

ho LeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct(s, a, r) (operation)

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

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 MorphismToLeftBidual (for IsCapCategoryObject)

▷ MorphismToLeftBidual(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 MorphismToLeftBidualWithGivenLeftBidual (for IsCapCategoryObject, Is-CapCategoryObject)

ightharpoonup MorphismToLeftBidualWithGivenLeftBidual(a, r)

(operation)

Returns: a morphism in Hom(a, r).

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 TensorProductLeftInternalHomCompatibilityMorphism (for IsList)

(operation)

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

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

1.5.24 TensorProductLeftInternalHomCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

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

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

TensorProductLeftInternalHomCompatibilityMorphismWithGivenObjects $_{a,a',b,b'}$: $\underline{\operatorname{Hom}}_{\ell}(a,a') \otimes \underline{\operatorname{Hom}}_{\ell}(b,b') \to \underline{\operatorname{Hom}}_{\ell}(a \otimes b,a' \otimes b')$.

1.5.25 TensorProductLeftDualityCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ TensorProductLeftDualityCompatibilityMorphism(a, b)

(operation)

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 TensorProductLeftDualityCompatibilityMorphism : $a^{\vee} \otimes b^{\vee} \to (a \otimes b)^{\vee}$.

1.5.26 TensorProductLeftDualityCompatibilityMorphismWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright TensorProductLeftDualityCompatibilityMorphismWithGivenObjects(s, a, b, r) (operation)

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

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

1.5.27 MorphismFromTensorProductToLeftInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromTensorProductToLeftInternalHom(a, b)

(operation)

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

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

1.5.28 MorphismFromTensorProductToLeftInternalHomWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MorphismFromTensorProductToLeftInternalHomWithGivenObjects(s, a, b, r) (operation)

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

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

1.5.29 IsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit (for Is-CapCategoryObject)

 $\verb| IsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit(a) | (attribute) \\ \textbf{Returns:} \ \ a \ morphism \ in \ Hom(a^{\vee}, \underline{Hom}_{\ell}(a,1)).$

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

1.5.30 IsomorphismFromLeftInternalHomIntoTensorUnitToLeftDualObject (for Is-CapCategoryObject)

▷ IsomorphismFromLeftInternalHomIntoTensorUnitToLeftDualObject(a) (attribute) **Returns:** a morphism in $\text{Hom}(\text{Hom}_{\ell}(a,1), a^{\vee})$.

The argument is an object a. The output is the isomorphism IsomorphismFromLeftInternalHomIntoTensorUnitToLeftDualObject_a: Hom_{ℓ} $(a, 1) \rightarrow a^{\vee}$.

1.5.31 UniversalPropertyOfLeftDual (for IsCapCategoryObject, IsCapCategoryMorphism)

▷ UniversalPropertyOfLeftDual(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 LeftClosedMonoidalLambdaIntroduction (for IsCapCategoryMorphism)

(attribute)

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

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

1.5.33 LeftClosedMonoidalLambdaElimination (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

▷ LeftClosedMonoidalLambdaElimination(a, b, alpha)

(operation)

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

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

1.5.34 IsomorphismFromObjectToLeftInternalHom (for IsCapCategoryObject)

▷ IsomorphismFromObjectToLeftInternalHom(a)

(attribute)

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

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

1.5.35 IsomorphismFromObjectToLeftInternalHomWithGivenLeftInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

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

1.5.36 IsomorphismFromLeftInternalHomToObject (for IsCapCategoryObject)

▷ IsomorphismFromLeftInternalHomToObject(a)

(attribute)

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

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

1.5.37 IsomorphismFromLeftInternalHomToObjectWithGivenLeftInternalHom (for IsCapCategoryObject, IsCapCategoryObject)

 \triangleright IsomorphismFromLeftInternalHomToObjectWithGivenLeftInternalHom(a, s) (operation) **Returns:** a morphism in Hom(s,a).

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

1.6 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 $\underline{\mathrm{Hom}}_{\ell}(b,-)$) is called a *closed monoidal category*.

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

The corresponding GAP property is called IsClosedMonoidalCategory.

1.6.1 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).

1.6.2 InternalHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ InternalHomOnMorphisms(alpha, beta)

(operation)

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

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

1.6.3 InternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

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

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

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 = Hom(a,b'). The output is the internal hom morphism $\text{Hom}(\alpha,\beta) : \text{Hom}(a',b) \to \text{Hom}(a,b')$.

1.6.4 ClosedMonoidalRightEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

 ${\tt \vartriangleright ClosedMonoidalRightEvaluationMorphism(a, b)}\\$

(operation)

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

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

1.6.5 ClosedMonoidalRightEvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject)

 \triangleright ClosedMonoidalRightEvaluationMorphismWithGivenSource(a, b, s) (operation) **Returns:** a morphism in Hom(s,b).

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

1.6.6 ClosedMonoidalRightCoevaluationMorphism (for IsCapCategoryObject, Is-CapCategoryObject)

▷ ClosedMonoidalRightCoevaluationMorphism(a, b)

(operation)

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

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

1.6.7 ClosedMonoidalRightCoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho ClosedMonoidalRightCoevaluationMorphismWithGivenRange(a, b, r) (operation) **Returns:** a morphism in Hom(b,r).

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

1.6.8 TensorProductToInternalHomRightAdjunctMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

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

1.6.9 TensorProductToInternalHomRightAdjunctMorphismWithGivenInternalHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject) IsCapCategoryObject)

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

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

1.6.10 TensorProductToInternalHomRightAdjunctionIsomorphism (for IsCapCategoryObject, IsCapCategoryObject)

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

The arguments are three objects a,b,c. The output is the tri-natural isomorphism $H(a \otimes b,c) \to H(b,\operatorname{Hom}(a,c))$ in the range category of the homomorphism structure H.

1.6.11 TensorProductToInternalHomRightAdjunctionIsomorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

The arguments are fives objects s, a, b, c, r where $s = H(a \otimes b, c)$ and $r = H(b, \underline{\text{Hom}}(a, c))$. The output is the tri-natural isomorphism $s \to r$ in the range category of the homomorphism structure H.

1.6.12 InternalHomToTensorProductRightAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

ightharpoonup InternalHomToTensorProductRightAdjunctMorphism(a, c, g) (operation) **Returns:** a morphism in Hom($a \otimes b, c$).

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

1.6.13 InternalHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| District | The transfer of the product Right Adjunct Morphism With Given Tensor Product (a, c, g, s) \\ (operation)$

Returns: a morphism in Hom(s,c).

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

1.6.14 InternalHomToTensorProductRightAdjunctionIsomorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ InternalHomToTensorProductRightAdjunctionIsomorphism(a, b, c) (operation)

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

The arguments are three objects a,b,c. The output is the tri-natural isomorphism $H(b, \text{Hom}(a,c)) \to H(a \otimes b,c)$ in the range category of the homomorphism structure H.

1.6.15 InternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ InternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects(s, a,
b, c, r)

(operation)

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

The arguments are fives objects s, a, b, c, r where $s = H(b, \underline{\text{Hom}}(a, c))$ and $r = H(a \otimes b, c)$. The output is the tri-natural isomorphism $s \to r$ in the range category of the homomorphism structure H.

1.6.16 ClosedMonoidalLeftEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ ClosedMonoidalLeftEvaluationMorphism(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 left evaluation morphism $\operatorname{ev}_{a,b}: \operatorname{\underline{Hom}}(a,b)\otimes a\to b$, i.e., the counit of the tensor hom adjunction.

1.6.17 ClosedMonoidalLeftEvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject)

 \triangleright ClosedMonoidalLeftEvaluationMorphismWithGivenSource(a, b, s) (operation) **Returns:** a morphism in Hom(s,b).

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

1.6.18 ClosedMonoidalLeftCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

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

(operation)

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

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

1.6.19 ClosedMonoidalLeftCoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup ClosedMonoidalLeftCoevaluationMorphismWithGivenRange(a, b, r) (operation) **Returns:** a morphism in $\operatorname{Hom}(b,r)$.

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

1.6.20 TensorProductToInternalHomLeftAdjunctMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

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

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

1.6.21 TensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 ${\tt \begin{tabular}{l} $ \vdash$ TensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom(a, b, f, i) \end{tabular} } \begin{tabular}{l} $ \vdash$ Operation \end{tabular}$

Returns: a morphism in Hom(a, i).

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 $g : a \to i$ corresponding to f under the tensor hom adjunction.

1.6.22 TensorProductToInternalHomLeftAdjunctionIsomorphism (for IsCapCategoryObject, IsCapCategoryObject)

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

The arguments are three objects a,b,c. The output is the tri-natural isomorphism $H(a \otimes b,c) \to H(a,\underline{\mathrm{Hom}}(b,c))$ in the range category of the homomorphism structure H.

1.6.23 TensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright TensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r) (operation)

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

The arguments are fives objects s, a, b, c, r where $s = H(a \otimes b, c)$ and $r = H(a, \underline{\text{Hom}}(b, c))$. The output is the tri-natural isomorphism $s \to r$ in the range category of the homomorphism structure H.

1.6.24 InternalHomToTensorProductLeftAdjunctMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism)

ightharpoonup InternalHomToTensorProductLeftAdjunctMorphism(b, c, g) (operation) **Returns:** a morphism in 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.

1.6.25 InternalHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\begin{tabular}{l} \rhd Internal Hom To Tensor Product Left Adjunct Morphism With Given Tensor Product (b, c, g, \\ s) & (operation) \\ \end{tabular}$

Returns: a morphism in Hom(s, c).

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

1.6.26 InternalHomToTensorProductLeftAdjunctionIsomorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\verb| Internal HomToTensor Product Left Adjunction Isomorphism (a, b, c) \\ \textbf{Returns:} \ \ a \ morphism \ in \ Hom(H(a,\underline{Hom}(b,c)),H(a\otimes b,c)). \end{aligned}$

The arguments are three objects a,b,c. The output is the tri-natural isomorphism $H(a,\underline{\operatorname{Hom}}(b,c)) \to H(a \otimes b,c)$ in the range category of the homomorphism structure H.

1.6.27 InternalHomToTensorProductLeftAdjunctionIsomorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright InternalHomToTensorProductLeftAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r) (operation)

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

The arguments are fives objects s, a, b, c, r where $s = H(a, \underline{\text{Hom}}(b, c))$ and $r = H(a \otimes b, c)$. The output is the tri-natural isomorphism $s \to r$ in the range category of the homomorphism structure H.

1.6.28 MonoidalPreComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ MonoidalPreComposeMorphism(a, b, c)

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

1.6.29 MonoidalPreComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MonoidalPreComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in Hom(s,r).

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

1.6.30 MonoidalPostComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject) Object, IsCapCategoryObject)

▷ MonoidalPostComposeMorphism(a, b, c)

(operation)

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

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

1.6.31 MonoidalPostComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\qquad \qquad \qquad \triangleright \ \, \texttt{MonoidalPostComposeMorphismWithGivenObjects}(\textit{s}, \ \textit{a, b, c, r}) \qquad \qquad (\textit{operation})$

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

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

1.6.32 DualOnObjects (for IsCapCategoryObject)

▷ DualOnObjects(a)

(attribute)

Returns: an object

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

1.6.33 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 $\alpha^{\vee}: b^{\vee} \to a^{\vee}$.

1.6.34 DualOnMorphismsWithGivenDuals (for IsCapCategoryObject, IsCapCategoryObject) ryMorphism, IsCapCategoryObject)

DualOnMorphismsWithGivenDuals(s, alpha, r)

(operation)

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

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 $\alpha^\vee:b^\vee\to a^\vee$.

1.6.35 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 $ev_a : a^{\vee} \otimes a \to 1$.

1.6.36 EvaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 ${\tt \vartriangleright} \ \, {\tt EvaluationForDualWithGivenTensorProduct}(s,\ a,\ r)$

(operation)

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

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

1.6.37 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 $a \to (a^{\vee})^{\vee}$.

1.6.38 MorphismToBidualWithGivenBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismToBidualWithGivenBidual(a, r)

(operation)

Returns: a morphism in Hom(a, r).

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

1.6.39 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 TensorProductInternalHomCompatibilityMorphism $_{a,a',b,b'}: \underline{\mathrm{Hom}}(a,a')\otimes \underline{\mathrm{Hom}}(b,b')\to \underline{\mathrm{Hom}}(a\otimes b,a'\otimes b')$.

1.6.40 TensorProductInternalHomCompatibilityMorphismWithGivenObjects (for Is-CapCategoryObject, IsList, IsCapCategoryObject)

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

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

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

1.6.41 TensorProductDualityCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject)

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

(operation)

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

1.6.42 TensorProductDualityCompatibilityMorphismWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup TensorProductDualityCompatibilityMorphismWithGivenObjects(s, a, b, r) (operation) Returns: a morphism in $\operatorname{Hom}(s,r)$.

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}: $a^{\vee} \otimes b^{\vee} \to (a \otimes b)^{\vee}$.

1.6.43 MorphismFromTensorProductToInternalHom (for IsCapCategoryObject, Is-CapCategoryObject)

 ${\tt \, \, \, \, \, \, MorphismFromTensorProductToInternalHom(a, \, \, b)}$

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

1.6.44 MorphismFromTensorProductToInternalHomWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MorphismFromTensorProductToInternalHomWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in Hom(s, r).

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

1.6.45 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 $_a: a^{\vee} \to \underline{\mathrm{Hom}}(a,1)$.

1.6.46 IsomorphismFromInternalHomIntoTensorUnitToDualObject (for IsCapCategoryObject)

(attribute)

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

The argument is an object a. The output is the isomorphism Isomorphism From Internal Hom Into Tensor Unit To Dual Object $a: \underline{\mathrm{Hom}}(a,1) \to a^{\vee}$.

1.6.47 UniversalPropertyOfDual (for IsCapCategoryObject, IsCapCategoryMorphism)

 \triangleright 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} .

1.6.48 LambdaIntroduction (for IsCapCategoryMorphism)

▷ LambdaIntroduction(alpha)

(attribute)

Returns: a morphism in Hom(1, 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.

1.6.49 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 α under the tensor hom adjunction.

1.6.50 IsomorphismFromObjectToInternalHom (for IsCapCategoryObject)

▷ IsomorphismFromObjectToInternalHom(a)

(attribute)

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

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

1.6.51 IsomorphismFromObjectToInternalHomWithGivenInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromObjectToInternalHomWithGivenInternalHom(a, r) (operation)

Returns: a morphism in Hom(a, r).

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

1.6.52 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 $\underline{\text{Hom}}(1,a) \to a$.

1.6.53 IsomorphismFromInternalHomToObjectWithGivenInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

 \triangleright IsomorphismFromInternalHomToObjectWithGivenInternalHom(a, s) (operation) **Returns:** a morphism in Hom(s, a).

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

1.7 Left 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 $\underline{\operatorname{coHom}}(-,b)$) is called a *left coclosed monoidal category*.

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

The corresponding GAP property is called IsLeftCoclosedMonoidalCategory.

1.7.1 LeftInternalCoHomOnObjects (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftInternalCoHomOnObjects(a, b)

(operation)

Returns: an object

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

1.7.2 LeftInternalCoHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

▷ LeftInternalCoHomOnMorphisms(alpha, beta)

(operation)

Returns: a morphism in $\text{Hom}(\underline{\text{coHom}}_{\ell}(a,b'),\underline{\text{coHom}}_{\ell}(a',b))$

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

1.7.3 LeftInternalCoHomOnMorphismsWithGivenLeftInternalCoHoms (for IsCap-CategoryObject, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategoryObject)

▷ LeftInternalCoHomOnMorphismsWithGivenLeftInternalCoHoms(s, alpha, beta, r)

Returns: a morphism in Hom(s, r)

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

1.7.4 LeftCoclosedMonoidalEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup LeftCoclosedMonoidalEvaluationMorphism(a, b)

(operation)

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

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

1.7.5 LeftCoclosedMonoidalEvaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup LeftCoclosedMonoidalEvaluationMorphismWithGivenRange(a, b, r) (operation) **Returns:** a morphism in Hom(b,r).

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

1.7.6 LeftCoclosedMonoidalCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

 $\qquad \qquad \triangleright \ \, \mathsf{LeftCoclosedMonoidalCoevaluationMorphism}(a,\ b)$

(operation)

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

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

1.7.7 LeftCoclosedMonoidalCoevaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftCoclosedMonoidalCoevaluationMorphismWithGivenSource(a, b, s) (operation) **Returns:** a morphism in Hom(s,b).

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

1.7.8 TensorProductToLeftInternalCoHomAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

▷ TensorProductToLeftInternalCoHomAdjunctMorphism(b, c, g) (operation)

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

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

1.7.9 TensorProductToLeftInternalCoHomAdjunctMorphismWithGivenLeftInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| TensorProductToLeftInternalCoHomAdjunctMorphismWithGivenLeftInternalCoHom(b, c, g, i) \\ (operation of the context of the$

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

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

1.7.10 LeftInternalCoHomToTensorProductAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

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

1.7.11 LeftInternalCoHomToTensorProductAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| LeftInternalCoHomToTensorProductAdjunctMorphismWithGivenTensorProduct(a, c, f, t) | (operation to be a substitution of the context of t$

Returns: a morphism in Hom(a,t).

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

1.7.12 LeftCoclosedMonoidalPreCoComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftCoclosedMonoidalPreCoComposeMorphism(a, b, c)

(operation)

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

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

1.7.13 LeftCoclosedMonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftCoclosedMonoidalPreCoComposeMorphismWithGivenObjects(s, a, b, c, r) (operation)

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

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

1.7.14 LeftCoclosedMonoidalPostCoComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup LeftCoclosedMonoidalPostCoComposeMorphism(a, b, c)

(operation)

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

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

1.7.15 LeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright LeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects(s, a, b, c, r) (operation)

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

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

1.7.16 LeftCoDualOnObjects (for IsCapCategoryObject)

▷ LeftCoDualOnObjects(a)

(attribute)

Returns: an object

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

1.7.17 LeftCoDualOnMorphisms (for IsCapCategoryMorphism)

▷ LeftCoDualOnMorphisms(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 $\alpha_{\vee}: b_{\vee} \to a_{\vee}$.

1.7.18 LeftCoDualOnMorphismsWithGivenLeftCoDuals (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 ${\tt \vartriangleright LeftCoDualOnMorphismsWithGivenLeftCoDuals(s, alpha, r)}\\$

(operation)

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

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 $\alpha_{\vee} : b^{\vee} \to a^{\vee}$.

1.7.19 LeftCoclosedMonoidalEvaluationForLeftCoDual (for IsCapCategoryObject)

▷ LeftCoclosedMonoidalEvaluationForLeftCoDual(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 $\operatorname{coclev}_a: 1 \to a_{\vee} \otimes a$.

1.7.20 LeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup LeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct(s, a, r)

(operation)

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

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 $\operatorname{coclev}_a: 1 \to a_{\vee} \otimes a$.

1.7.21 MorphismFromLeftCoBidual (for IsCapCategoryObject)

▷ MorphismFromLeftCoBidual(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 $(a_{\vee})_{\vee} \to a$.

1.7.22 MorphismFromLeftCoBidualWithGivenLeftCoBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromLeftCoBidualWithGivenLeftCoBidual(a, s)

(operation)

Returns: a morphism in Hom(s, a).

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

1.7.23 LeftInternalCoHomTensorProductCompatibilityMorphism (for IsList)

 $\qquad \qquad \triangleright \ \, \mathsf{LeftInternalCoHomTensorProductCompatibilityMorphism}(\mathit{list})$

(operation)

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

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

1.7.24 LeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

 $\verb| LeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects(s, list, r) | (operation) | (operation)$

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

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

1.7.25 LeftCoDualityTensorProductCompatibilityMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ LeftCoDualityTensorProductCompatibilityMorphism(a, b)

(operation)

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

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

1.7.26 LeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\verb| LeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects(s, a, b, r)| \\ (operation)$

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

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 LeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects $_{a,b}:(a\otimes b)_\vee\to a_\vee\otimes b_\vee$.

1.7.27 MorphismFromLeftInternalCoHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromLeftInternalCoHomToTensorProduct(a, b)

(operation)

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

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

1.7.28 MorphismFromLeftInternalCoHomToTensorProductWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MorphismFromLeftInternalCoHomToTensorProductWithGivenObjects(s, a, b, r) (operation)

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

The arguments object $coHom_{\ell}(a,b)$, a, b,The object output is the natural morphism and an $r = b_{\vee} \otimes a$. ${\bf MorphismFromLeftInternalCoHomToTensorProductWithGivenObjects}_{a.b}: \underline{{\bf coHom}}_{\ell}(a,b) \rightarrow a \otimes b_{\vee}.$

1.7.29 IsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit (for IsCapCategoryObject)

ightharpoonup IsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit(a) (attribute) **Returns:** a morphism in $\operatorname{Hom}(a_{\vee}, \operatorname{\underline{coHom}}_{\ell}(1, a))$.

The argument is an object a. The output is the isomorphism IsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit_a: $a_{\lor} \to \underline{\text{coHom}}_{\ell}(1, a)$.

1.7.30 IsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject (for IsCapCategoryObject)

▷ IsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject(a) (attribute)

Returns: a morphism in $Hom(coHom_{\ell}(1,a),a_{\vee})$.

The argument is an object a. The output is the isomorphism IsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject_a: $coHom_{\ell}(1,a) \rightarrow a_{\vee}$.

1.7.31 UniversalPropertyOfLeftCoDual (for IsCapCategoryObject, IsCapCategoryMorphism)

▷ UniversalPropertyOfLeftCoDual(t, a, alpha)

(operation)

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

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} .

1.7.32 LeftCoclosedMonoidalLambdaIntroduction (for IsCapCategoryMorphism)

 ${\tt \vartriangleright LeftCoclosedMonoidalLambdaIntroduction(alpha)}$

(attribute)

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

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

1.7.33 LeftCoclosedMonoidalLambdaElimination (for IsCapCategoryObject, IsCapCategoryMorphism)

 ${\tt \vartriangleright LeftCoclosedMonoidalLambdaElimination(a, b, alpha)}\\$

(operation)

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

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

1.7.34 IsomorphismFromObjectToLeftInternalCoHom (for IsCapCategoryObject)

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

(attribute)

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

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

1.7.35 IsomorphismFromObjectToLeftInternalCoHomWithGivenLeftInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject)

 \triangleright IsomorphismFromObjectToLeftInternalCoHomWithGivenLeftInternalCoHom(a, r) (operation)

Returns: a morphism in Hom(a, r).

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

1.7.36 IsomorphismFromLeftInternalCoHomToObject (for IsCapCategoryObject)

▷ IsomorphismFromLeftInternalCoHomToObject(a)

(attribute)

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

The argument is an object a. The output is the natural isomorphism $coHom_{\ell}(a,1) \rightarrow a$.

1.7.37 IsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject)

 \triangleright IsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom(a, s) (operation)

Returns: a morphism in Hom(s, a).

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

1.8 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 $\underline{\operatorname{coHom}}(-,b)$) is called a *coclosed monoidal category*.

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

The corresponding GAP property is called IsCoclosedMonoidalCategory.

1.8.1 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 coHom(a, b).

1.8.2 InternalCoHomOnMorphisms (for IsCapCategoryMorphism, IsCapCategoryMorphism)

 \triangleright 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 $\operatorname{\underline{coHom}}(\alpha,\beta): \operatorname{\underline{coHom}}(a,b') \to \operatorname{\underline{coHom}}(a',b)$.

1.8.3 InternalCoHomOnMorphismsWithGivenInternalCoHoms (for IsCapCategory-Object, IsCapCategoryMorphism, IsCapCategoryMorphism, IsCapCategory-Object)

 \triangleright InternalCoHomOnMorphismsWithGivenInternalCoHoms(s, alpha, beta, r) (operation) **Returns:** a morphism in Hom(s,r)

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 $\underline{\operatorname{coHom}}(\alpha,\beta) : \underline{\operatorname{coHom}}(a,b') \to \operatorname{coHom}(a',b)$.

1.8.4 CoclosedMonoidalRightEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedMonoidalRightEvaluationMorphism(a, b)

(operation)

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

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

1.8.5 CoclosedMonoidalRightEvaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup CoclosedMonoidalRightEvaluationMorphismWithGivenRange(a, b, r) (operation) **Returns:** a morphism in $\operatorname{Hom}(b,r)$.

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

1.8.6 CoclosedMonoidalRightCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

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

(operation)

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

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

1.8.7 CoclosedMonoidalRightCoevaluationMorphismWithGivenSource (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 $\verb| CoclosedMonoidalRightCoevaluationMorphismWithGivenSource(a, b, s) \\ \textbf{Returns:} \ \ \text{a morphism in } \text{Hom}(s,b).$

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

1.8.8 TensorProductToInternalCoHomRightAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

 \triangleright TensorProductToInternalCoHomRightAdjunctMorphism(b, c, g) (operation) **Returns:** a morphism in Hom(coHom(a,b),c).

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

1.8.9 TensorProductToInternalCoHomRightAdjunctMorphismWithGivenInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| TensorProductToInternalCoHomRightAdjunctMorphismWithGivenInternalCoHom(b, c, g, i) \\ (operation) \\$

Returns: a morphism in Hom(i, c).

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

1.8.10 InternalCoHomToTensorProductRightAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

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

1.8.11 InternalCoHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| InternalCoHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct(a, b, f, t) | (operation) | (operatio$

Returns: a morphism in Hom(a,t).

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

1.8.12 CoclosedMonoidalLeftEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedMonoidalLeftEvaluationMorphism(a, b)

(operation)

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

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

1.8.13 CoclosedMonoidalLeftEvaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject)

ightharpoonup CoclosedMonoidalLeftEvaluationMorphismWithGivenRange(a, b, r) (operation) **Returns:** a morphism in $\operatorname{Hom}(b,r)$.

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

1.8.14 CoclosedMonoidalLeftCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedMonoidalLeftCoevaluationMorphism(a, b)

(operation)

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

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

1.8.15 CoclosedMonoidalLeftCoevaluationMorphismWithGivenSource (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright CoclosedMonoidalLeftCoevaluationMorphismWithGivenSource(a, b, s) (operation) **Returns:** a morphism in Hom(s,b).

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

1.8.16 TensorProductToInternalCoHomLeftAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

 \triangleright TensorProductToInternalCoHomLeftAdjunctMorphism(b, c, g) (operation) **Returns:** a morphism in Hom(coHom(a,c),b).

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

1.8.17 TensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

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

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

1.8.18 InternalCoHomToTensorProductLeftAdjunctMorphism (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryMorphism)

ightharpoonup InternalCoHomToTensorProductLeftAdjunctMorphism(a, c, f) (operation) **Returns:** a morphism in Hom(a, b⊗c).

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

1.8.19 InternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 $\verb| > InternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(a, c, f, t) \\ (operation)$

Returns: a morphism in Hom(a,t).

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

1.8.20 MonoidalPreCoComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

1.8.21 MonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MonoidalPreCoComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in Hom(s,r).

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

1.8.22 MonoidalPostCoComposeMorphism (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ MonoidalPostCoComposeMorphism(a, b, c)

(operation)

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

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

1.8.23 MonoidalPostCoComposeMorphismWithGivenObjects (for IsCapCategory-Object, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MonoidalPostCoComposeMorphismWithGivenObjects(s, a, b, c, r) (operation) **Returns:** a morphism in $\operatorname{Hom}(s,r)$.

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

1.8.24 CoDualOnObjects (for IsCapCategoryObject)

▷ CoDualOnObjects(a)

(attribute)

Returns: an object

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

1.8.25 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 $\alpha_{\vee}: b_{\vee} \to a_{\vee}$.

1.8.26 CoDualOnMorphismsWithGivenCoDuals (for IsCapCategoryObject, IsCapCategoryMorphism, IsCapCategoryObject)

 \triangleright CoDualOnMorphismsWithGivenCoDuals(s, alpha, r)

(operation)

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

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 $\alpha_{\vee} : b^{\vee} \to a^{\vee}$.

1.8.27 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 $\operatorname{coclev}_a: 1 \to a_{\vee} \otimes a$.

1.8.28 CoclosedEvaluationForCoDualWithGivenTensorProduct (for IsCapCategory-Object, IsCapCategoryObject)

riangleright CoclosedEvaluationForCoDualWithGivenTensorProduct $(s,\ a,\ r)$

(operation)

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

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 $\operatorname{coclev}_a: 1\to a_\vee\otimes a$.

1.8.29 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 $(a_{\vee})_{\vee} \to a$.

1.8.30 MorphismFromCoBidualWithGivenCoBidual (for IsCapCategoryObject, Is-CapCategoryObject)

(operation)

Returns: a morphism in Hom(s, a).

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

1.8.31 InternalCoHomTensorProductCompatibilityMorphism (for IsList)

▷ 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 InternalCoHomTensorProductCompatibilityMorphism $_{a,a',b,b'}$: $\underline{\operatorname{coHom}}(a\otimes a',b\otimes b')\to \underline{\operatorname{coHom}}(a,b)\otimes\underline{\operatorname{coHom}}(a',b')$.

1.8.32 InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

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

(operation)

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

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

1.8.33 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 CoDualityTensorProductCompatibilityMorphism : $(a \otimes b)_{\vee} \to a_{\vee} \otimes b_{\vee}$.

1.8.34 CoDualityTensorProductCompatibilityMorphismWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

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

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_{a,b}: $(a\otimes b)_{\vee}\to a_{\vee}\otimes b_{\vee}$.

1.8.35 MorphismFromInternalCoHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromInternalCoHomToTensorProduct(a, b)

(operation)

Returns: a morphism in Hom(coHom(a,b),b $\lor \otimes a$).

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

1.8.36 MorphismFromInternalCoHomToTensorProductWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

 \triangleright MorphismFromInternalCoHomToTensorProductWithGivenObjects(s, a, b, r) (operation) **Returns:** a morphism in Hom(s, r).

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_{a,b}: $\operatorname{coHom}(a,b) \to a \otimes b_{\vee}$.

1.8.37 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 $_a: a_{\lor} \to \underline{\mathrm{coHom}}(1,a)$.

1.8.38 IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject (for Is-CapCategoryObject)

▷ IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject(a) (attribute)

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

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

1.8.39 UniversalPropertyOfCoDual (for 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} .

1.8.40 CoLambdaIntroduction (for IsCapCategoryMorphism)

▷ CoLambdaIntroduction(alpha)

(attribute)

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

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.

1.8.41 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 α under the cohom tensor adjunction.

1.8.42 IsomorphismFromObjectToInternalCoHom (for IsCapCategoryObject)

▷ 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 $a \to \text{coHom}(a, 1)$.

1.8.43 IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom (for Is-CapCategoryObject, IsCapCategoryObject)

ightharpoonup IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom(a, r) (operation) **Returns:** a morphism in Hom(a, r).

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

1.8.44 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 $\underline{\text{coHom}}(a, 1) \to a$.

1.8.45 IsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom (for Is-CapCategoryObject, IsCapCategoryObject)

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

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

1.9 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.

1.10 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.

1.11 Rigid Symmetric Closed Monoidal Categories

A symmetric closed monoidal category C satisfying

• the natural morphism

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

• the natural morphism

 $a \to \underline{\operatorname{Hom}}_{\ell}(\underline{\operatorname{Hom}}_{\ell}(a,1),1)$ 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 $\underline{\operatorname{Hom}}_{\ell}(a,b) \coloneqq a^{\vee} \otimes b$ and, in particular, $\underline{\operatorname{Hom}}_{\ell}(a,1) \coloneqq a^{\vee} \otimes 1$.

The corresponding GAP property is given by IsRigidSymmetricClosedMonoidalCategory.

1.11.1 IsomorphismFromTensorProductWithDualObjectToInternalHom (for IsCap-CategoryObject, IsCapCategoryObject)

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

1.11.2 IsomorphismFromInternalHomToTensorProductWithDualObject (for IsCap-CategoryObject, IsCapCategoryObject)

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

1.11.3 MorphismFromInternalHomToTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

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

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

1.11.4 MorphismFromInternalHomToTensorProductWithGivenObjects (for IsCap-CategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

1.11.5 TensorProductInternalHomCompatibilityMorphismInverse (for IsList)

 ${\tt \vartriangleright} \verb| TensorProductInternalHomCompatibilityMorphismInverse(| list)|$

(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 $_{a,a',b,b'}$: $\underline{\text{Hom}}(a \otimes b,a' \otimes b') \to \underline{\text{Hom}}(a,a') \otimes \underline{\text{Hom}}(b,b')$.

1.11.6 TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

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

1.11.7 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 $coev_a: 1 \to a \otimes a^{\vee}$.

1.11.8 CoevaluationForDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject) CapCategoryObject, IsCapCategoryObject)

 ${\tt \vartriangleright} \ \, {\tt CoevaluationForDualWithGivenTensorProduct}(s,\ a,\ r)\\$

(operation)

Returns: a morphism in 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 $\operatorname{coev}_a: 1\to a\otimes a^{\vee}$.

1.11.9 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_{\alpha}: $1 \to 1$.

1.11.10 RankMorphism (for IsCapCategoryObject)

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

The argument is an object a. The output is the rank morphism rank_a: $1 \rightarrow 1$.

1.11.11 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 $(a^{\vee})^{\vee} \to a$.

1.11.12 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 $(a^{\vee})^{\vee} \to a$.

1.12 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.

1.12.1 IsomorphismFromInternalCoHomToTensorProductWithCoDualObject (for IsCapCategoryObject, IsCapCategoryObject)

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

1.12.2 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 $_{a,b}: a_{\vee}\otimes b \to \underline{\mathrm{coHom}}(b,a)$.

1.12.3 MorphismFromTensorProductToInternalCoHom (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismFromTensorProductToInternalCoHom(a, b)

(operation)

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

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

1.12.4 MorphismFromTensorProductToInternalCoHomWithGivenObjects (for Is-CapCategoryObject, IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

ho 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_{a,b}: $a_{\vee} \otimes b \to \underline{\operatorname{coHom}}(b,a)$.

1.12.5 InternalCoHomTensorProductCompatibilityMorphismInverse (for IsList)

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

1.12.6 InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects (for IsCapCategoryObject, IsList, IsCapCategoryObject)

 $\verb| InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects(s, list, r) \\ \\ (operation)$

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

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

1.12.7 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 $\operatorname{coclcoev}_a: a \otimes a_{\vee} \to 1$.

1.12.8 CoclosedCoevaluationForCoDualWithGivenTensorProduct (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedCoevaluationForCoDualWithGivenTensorProduct(s, a, r)

(operation)

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

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 $\operatorname{coclcoev}_a : a \otimes a_{\vee} \to 1$.

1.12.9 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 $\operatorname{cotrace}_{\alpha}: 1 \to 1$.

1.12.10 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_a: $1 \rightarrow 1$.

1.12.11 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 $a \to (a_{\vee})_{\vee}$.

1.12.12 MorphismToCoBidualWithGivenCoBidual (for IsCapCategoryObject, IsCapCategoryObject)

▷ MorphismToCoBidualWithGivenCoBidual(a, r)

(operation)

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

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

1.13 Convenience Methods

1.13.1 InternalHom (for IsCapCategoryCell, IsCapCategoryCell)

▷ 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 $\underline{\text{Hom}}(a,b)$. 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.

1.13.2 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.

1.13.3 LeftInternalHom (for IsCapCategoryCell, IsCapCategoryCell)

▷ LeftInternalHom(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 $\underline{\mathrm{Hom}}_{\ell}(a,b)$. 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.

1.13.4 LeftInternalCoHom (for IsCapCategoryCell, IsCapCategoryCell)

▷ LeftInternalCoHom(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 $\underline{\operatorname{coHom}}_{\ell}(a,b)$. 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.

1.14 Add-methods

1.14.1 AddLeftDistributivityExpanding (for IsCapCategory, IsFunction)

▷ AddLeftDistributivityExpanding(C, F)

(operation)

▷ AddLeftDistributivityExpanding(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational

complexity of the function (lower weight = less complex = faster execution). $F:(a,L) \mapsto \text{LeftDistributivityExpanding}(a,L)$.

1.14.2 AddLeftDistributivityExpandingWithGivenObjects (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,L,r)\mapsto \text{LeftDistributivityExpandingWithGivenObjects}(s,a,L,r).$

1.14.3 AddLeftDistributivityFactoring (for IsCapCategory, IsFunction)

```
▷ AddLeftDistributivityFactoring(C, F) (operation)
▷ AddLeftDistributivityFactoring(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,L)\mapsto \text{LeftDistributivityFactoring}(a,L)$.

1.14.4 AddLeftDistributivityFactoringWithGivenObjects (for IsCapCategory, IsFunction)

```
▷ AddLeftDistributivityFactoringWithGivenObjects(C, F) (operation)
▷ AddLeftDistributivityFactoringWithGivenObjects(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,L,r)\mapsto \text{LeftDistributivityFactoringWithGivenObjects}(s,a,L,r).$

1.14.5 AddRightDistributivityExpanding (for IsCapCategory, 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 RightDistributivityExpanding. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(L,a)\mapsto RightDistributivityExpanding(L,a)$.

1.14.6 AddRightDistributivityExpandingWithGivenObjects (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (s,L,a,r) \mapsto \text{RightDistributivityExpandingWithGivenObjects}(s,L,a,r)$.

1.14.7 AddRightDistributivityFactoring (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(L,a) \mapsto \text{RightDistributivityFactoring}(L,a)$.

1.14.8 AddRightDistributivityFactoringWithGivenObjects (for IsCapCategory, 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 RightDistributivityFactoringWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,L,a,r)\mapsto RightDistributivityFactoringWithGivenObjects(s,L,a,r)$.

1.14.9 AddBraiding (for IsCapCategory, IsFunction)

▷ AddBraiding(C, F) (operation)

▷ AddBraiding(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \text{Braiding}(a,b)$.

1.14.10 AddBraidingInverse (for IsCapCategory, 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 BraidingInverse. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \text{BraidingInverse}(a,b)$.

1.14.11 AddBraidingInverseWithGivenTensorProducts (for IsCapCategory, 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 BraidingInverseWithGivenTensorProducts. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto \text{BraidingInverseWithGivenTensorProducts}(s,a,b,r).$

1.14.12 AddBraidingWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddBraidingWithGivenTensorProducts(C, F) (operation)

▷ AddBraidingWithGivenTensorProducts(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto \text{BraidingWithGivenTensorProducts}(s,a,b,r).$

1.14.13 AddClosedMonoidalLeftCoevaluationMorphism (for IsCapCategory, 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 ClosedMonoidalLeftCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto ClosedMonoidalLeftCoevaluationMorphism(a,b)$.

1.14.14 AddClosedMonoidalLeftCoevaluationMorphismWithGivenRange (for Is-CapCategory, IsFunction)

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 ClosedMonoidalLeftCoevaluationMorphismWithGivenRange. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r)\mapsto ClosedMonoidalLeftCoevaluationMorphismWithGivenRange(<math>a,b,r$).

1.14.15 AddClosedMonoidalLeftEvaluationMorphism (for IsCapCategory, IsFunction)

 ${} \hspace*{0.2cm} \hspace$

▷ AddClosedMonoidalLeftEvaluationMorphism(C, F, weight)

(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 ClosedMonoidalLeftEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto ClosedMonoidalLeftEvaluationMorphism(a,b)$.

1.14.16 AddClosedMonoidalLeftEvaluationMorphismWithGivenSource (for IsCap-Category, IsFunction)

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

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 ClosedMonoidalLeftEvaluationMorphismWithGivenSource. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto ClosedMonoidalLeftEvaluationMorphismWithGivenSource(a,b,s)$.

1.14.17 AddClosedMonoidalRightCoevaluationMorphism (for IsCapCategory, IsFunction)

- ${} \hspace*{0.2cm} \hspace$
- ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}} {\tt AddClosedMonoidalRightCoevaluationMorphism}({\tt C, F, weight}) \\$

(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 ClosedMonoidalRightCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto ClosedMonoidalRightCoevaluationMorphism(a,b)$.

1.14.18 AddClosedMonoidalRightCoevaluationMorphismWithGivenRange (for Is-CapCategory, IsFunction)

 $\begin{tabular}{l} $ > $ AddClosedMonoidalRightCoevaluationMorphismWithGivenRange(C, F) & (operation) \\ $ > $ AddClosedMonoidalRightCoevaluationMorphismWithGivenRange(C, F, weight) & (operation) \\ \end{tabular}$

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 ClosedMonoidalRightCoevaluationMorphismWithGivenRange. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r)\mapsto ClosedMonoidalRightCoevaluationMorphismWithGivenRange(<math>a,b,r$).

1.14.19 AddClosedMonoidalRightEvaluationMorphism (for IsCapCategory, IsFunction)

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

▷ AddClosedMonoidalRightEvaluationMorphism(C, F, weight) (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 ClosedMonoidalRightEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto ClosedMonoidalRightEvaluationMorphism(a,b)$.

1.14.20 AddClosedMonoidalRightEvaluationMorphismWithGivenSource (for IsCap-Category, IsFunction)

▷ AddClosedMonoidalRightEvaluationMorphismWithGivenSource(C, F) (operation)
▷ AddClosedMonoidalRightEvaluationMorphismWithGivenSource(C, F, weight) (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 ClosedMonoidalRightEvaluationMorphismWithGivenSource. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto ClosedMonoidalRightEvaluationMorphismWithGivenSource(a,b,s).$

1.14.21 AddDualOnMorphisms (for IsCapCategory, IsFunction)

```
    AddDualOnMorphisms(C, F) (operation)

    AddDualOnMorphisms(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \text{DualOnMorphisms}(alpha)$.

1.14.22 AddDualOnMorphismsWithGivenDuals (for IsCapCategory, IsFunction)

▷ AddDualOnMorphismsWithGivenDuals(C, F)

(operation)

▷ AddDualOnMorphismsWithGivenDuals(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,r)\mapsto DualOnMorphismsWithGivenDuals(s,alpha,r)$.

1.14.23 AddDualOnObjects (for IsCapCategory, IsFunction)

▷ AddDualOnObjects(C, F)

(operation)

▷ AddDualOnObjects(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{DualOnObjects}(a)$.

1.14.24 AddEvaluationForDual (for IsCapCategory, IsFunction)

▷ AddEvaluationForDual(C, F)

(operation)

▷ AddEvaluationForDual(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{EvaluationForDual}(a)$.

1.14.25 AddEvaluationForDualWithGivenTensorProduct (for IsCapCategory, Is-Function)

 \triangleright AddEvaluationForDualWithGivenTensorProduct(C, F)

(operation)

▷ AddEvaluationForDualWithGivenTensorProduct(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto \text{EvaluationForDualWithGivenTensorProduct}(s,a,r)$.

1.14.26 AddInternalHomOnMorphisms (for IsCapCategory, IsFunction)

AddInternalHomOnMorphisms(C, F)

(operation)

▷ AddInternalHomOnMorphisms(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha,beta) \mapsto InternalHomOnMorphisms(alpha,beta)$.

1.14.27 AddInternalHomOnMorphismsWithGivenInternalHoms (for IsCapCategory, IsFunction)

- ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$
- ${\tt \triangleright} \ \, {\tt AddInternalHomOnMorphismsWithGivenInternalHoms}(\textit{C, F, weight}) \qquad \qquad ({\tt operation})$

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,beta,r)\mapsto$ InternalHomOnMorphismsWithGivenInternalHoms(s,alpha,beta,r).

1.14.28 AddInternalHomOnObjects (for IsCapCategory, IsFunction)

ightharpoonup AddInternalHomOnObjects(C, F) (operation) ightharpoonup (operation) ightharpoonup (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \text{InternalHomOnObjects}(a,b)$.

1.14.29 AddInternalHomToTensorProductLeftAdjunctMorphism (for IsCapCategory, IsFunction)

▷ AddInternalHomToTensorProductLeftAdjunctMorphism(C, F) (operation)
▷ AddInternalHomToTensorProductLeftAdjunctMorphism(C, F, weight) (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 InternalHomToTensorProductLeftAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g)\mapsto$ InternalHomToTensorProductLeftAdjunctMorphism(b,c,g).

1.14.30 AddInternalHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct (for IsCapCategory, IsFunction)

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

(operation)

▷ AddInternalHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(C, F,

weight) (operation)

Returns: nothing

F. The arguments category Cfunction This are and operabasic adds the given function Fto the category for the operation Internal Hom To Tensor Product Left Adjunct Morphism With Given Tensor Product.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g,s)\mapsto$ InternalHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(b, c, g, s).

1.14.31 AddInternalHomToTensorProductLeftAdjunctionIsomorphism (for IsCap-Category, IsFunction)

- ▷ AddInternalHomToTensorProductLeftAdjunctionIsomorphism(C, F) (operation)
- \triangleright AddInternalHomToTensorProductLeftAdjunctionIsomorphism(C, F, weight) (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 InternalHomToTensorProductLeftAdjunctionIsomorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto$ InternalHomToTensorProductLeftAdjunctionIsomorphism(a,b,c).

1.14.32 AddInternalHomToTensorProductLeftAdjunctionIsomorphismWithGivenObjects (for IsCapCategory, IsFunction)

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

(operation)

 $\begin{tabular}{l} $ > $ AddInternal HomToTensor Product Left Adjunction Isomorphism With Given Objects ({\it C, F, weight}) \\ \hline \end{tabular} $ (operation) $ (operation)$

Returns: nothing

F. The arguments are category Cand function This operaa the given function Fto the category for operation Internal Hom To Tensor Product Left Adjunction Isomorphism With Given Objects.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ InternalHomToTensorProductLeftAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r).

1.14.33 AddInternalHomToTensorProductRightAdjunctMorphism (for IsCapCategory, IsFunction)

- ${\tt \hspace*{0.5cm}} {\tt \hspace*{0.5cm}}$
- > AddInternalHomToTensorProductRightAdjunctMorphism(C, F, weight) (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 InternalHomToTensorProductRightAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,g)\mapsto InternalHomToTensorProductRightAdjunctMorphism(a,c,g)$.

1.14.34 AddInternalHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct (for IsCapCategory, IsFunction)

Returns: nothing

This The arguments function F. operaare category Cand adds the given function F the category the basic operation tion to for Internal HomToTensorProductRightAdjunctMorphismWithGivenTensorProduct. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,g,s)\mapsto$ InternalHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct(a, c, g, s).

1.14.35 AddInternalHomToTensorProductRightAdjunctionIsomorphism (for IsCap-Category, IsFunction)

▷ AddInternalHomToTensorProductRightAdjunctionIsomorphism(C, F) (operation)
▷ AddInternalHomToTensorProductRightAdjunctionIsomorphism(C, F, weight) (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 InternalHomToTensorProductRightAdjunctionIsomorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto$ InternalHomToTensorProductRightAdjunctionIsomorphism(a,b,c).

1.14.36 AddInternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ \land$ AddInternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects(C, F) & (operation) \\ $ \land$ AddInternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects(C, F, weight) & (operation) \\ \end{tabular}$

Returns: nothing

arguments F. This The category Cfunction operaare and given tion adds the function F to the category for the basic operation Internal HomToTensorProductRightAdjunctionIsomorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ InternalHomToTensorProductRightAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r).

1.14.37 AddIsomorphismFromDualObjectToInternalHomIntoTensorUnit (for IsCap-Category, IsFunction)

▷ AddIsomorphismFromDualObjectToInternalHomIntoTensorUnit(C, F) (operation)
▷ AddIsomorphismFromDualObjectToInternalHomIntoTensorUnit(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto IsomorphismFromDualObjectToInternalHomIntoTensorUnit(a)$.

1.14.38 AddIsomorphismFromInternalHomIntoTensorUnitToDualObject (for IsCap-Category, IsFunction)

- \triangleright AddIsomorphismFromInternalHomIntoTensorUnitToDualObject(C, F) (operation)
- ▷ AddIsomorphismFromInternalHomIntoTensorUnitToDualObject(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto IsomorphismFromInternalHomIntoTensorUnitToDualObject(a)$.

1.14.39 AddIsomorphismFromInternalHomToObject (for IsCapCategory, IsFunction)

- - **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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto IsomorphismFromInternalHomToObject(a)$.

1.14.40 AddIsomorphismFromInternalHomToObjectWithGivenInternalHom (for Is-CapCategory, IsFunction)

 \triangleright AddIsomorphismFromInternalHomToObjectWithGivenInternalHom(C, F) (operation) \triangleright AddIsomorphismFromInternalHomToObjectWithGivenInternalHom(C, F, weight) (operation)

Returns: nothing

The arguments CF. This are a category and function operathe given function Fto the category for the basic operation Isomorphism From Internal Hom ToObject With Given Internal Hom.Optionally, weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). IsomorphismFromInternalHomToObjectWithGivenInternalHom(a, s).

1.14.41 AddIsomorphismFromObjectToInternalHom (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto IsomorphismFromObjectToInternalHom(a)$.

1.14.42 AddIsomorphismFromObjectToInternalHomWithGivenInternalHom (for Is-CapCategory, IsFunction)

 $> AddIsomorphismFromObjectToInternalHomWithGivenInternalHom({\it C}, {\it F}) \qquad {\rm (operation)} \\ > AddIsomorphismFromObjectToInternalHomWithGivenInternalHom({\it C}, {\it F}, {\it weight}) \qquad {\rm (operation)}$

Returns: nothing

arguments F. This The Cfunction operaare category and function Fthe category operation adds the given to for the basic Isomorphism From Object To Internal Hom With Given Internal Hom.Optionally, weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). IsomorphismFromObjectToInternalHomWithGivenInternalHom(a, r).

1.14.43 AddLambdaElimination (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (a, b, alpha) \mapsto \text{LambdaElimination}(a, b, alpha)$.

1.14.44 AddLambdaIntroduction (for IsCapCategory, IsFunction)

▷ AddLambdaIntroduction(C, F) (operation)

▷ AddLambdaIntroduction(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(al\,pha)\mapsto \texttt{LambdaIntroduction}(al\,pha)$.

1.14.45 AddMonoidalPostComposeMorphism (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto MonoidalPostComposeMorphism(a,b,c)$.

1.14.46 AddMonoidalPostComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto MonoidalPostComposeMorphismWithGivenObjects(s,a,b,c,r)$.

1.14.47 AddMonoidalPreComposeMorphism (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \texttt{MonoidalPreComposeMorphism}(a,b,c)$.

1.14.48 AddMonoidalPreComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMonoidalPreComposeMorphismWithGivenObjects(C, F) (operation)

▷ AddMonoidalPreComposeMorphismWithGivenObjects(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto MonoidalPreComposeMorphismWithGivenObjects(s,a,b,c,r)$.

1.14.49 AddMorphismFromTensorProductToInternalHom (for IsCapCategory, Is-Function)

 ${\hspace{0.2cm}} {\hspace{0.2cm}} {\hspace{0.2cm}$

▷ AddMorphismFromTensorProductToInternalHom(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromTensorProductToInternalHom(a,b)$.

1.14.50 AddMorphismFromTensorProductToInternalHomWithGivenObjects (for Is-CapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddMorphismFromTensorProductToInternalHomWithGivenObjects({\it C}, {\it F}) $ & operation) \\ $ > $ AddMorphismFromTensorProductToInternalHomWithGivenObjects({\it C}, {\it F}, {\it weight}) $ & operation) \\ \end{tabular}$

Returns: nothing

arguments F. This The category Cand function operaare given function F the category the operation adds the to for basic ${\tt MorphismFromTensorProductToInternalHomWithGivenObjects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ MorphismFromTensorProductToInternalHomWithGivenObjects(s, a, b, r).

1.14.51 AddMorphismToBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToBidual(C, F)

(operation)

▷ AddMorphismToBidual(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{MorphismToBidual}(a)$.

1.14.52 AddMorphismToBidualWithGivenBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToBidualWithGivenBidual(C, F)

(operation)

▷ AddMorphismToBidualWithGivenBidual(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto MorphismToBidualWithGivenBidual(a,r)$.

1.14.53 AddTensorProductDualityCompatibilityMorphism (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto TensorProductDualityCompatibilityMorphism(a,b)$.

1.14.54 AddTensorProductDualityCompatibilityMorphismWithGivenObjects (for Is-CapCategory, IsFunction)

 \triangleright AddTensorProductDualityCompatibilityMorphismWithGivenObjects(C, F) (operation) \triangleright AddTensorProductDualityCompatibilityMorphismWithGivenObjects(C, F, weight)

(operation)

Returns: nothing

The arguments F. are category Cand function This operaa function adds the given Fto the category for the basic operation $Tensor Product Duality Compatibility Morphism With {\tt Given Objects}.$ Optionally, 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto \texttt{TensorProductDualityCompatibilityMorphismWithGivenObjects}(s,a,b,r).$

1.14.55 AddTensorProductInternalHomCompatibilityMorphism (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(list) \mapsto TensorProductInternalHomCompatibilityMorphism(list)$.

1.14.56 AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

 \triangleright AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects(C, F) (operation)

 $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductInternalHomCompatibilityMorphismWithGivenObjects}(\mathit{C},\ \mathit{F},\\ \mathit{weight}) \qquad \qquad \qquad (operation)$

Returns: nothing

F. The arguments Cfunction This operaare category and tion adds the given function F to the category for the basic operation

TensorProductInternalHomCompatibilityMorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(source, list, range) \mapsto TensorProductInternalHomCompatibilityMorphismWithGivenObjects(source, list, range)$.

1.14.57 AddTensorProductToInternalHomLeftAdjunctMorphism (for IsCapCategory, IsFunction)

- ${} \hspace*{0.2cm} \hspace$
- → AddTensorProductToInternalHomLeftAdjunctMorphism(C, F, weight) (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 TensorProductToInternalHomLeftAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f)\mapsto TensorProductToInternalHomLeftAdjunctMorphism(a,b,f)$.

1.14.58 AddTensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom (for IsCapCategory, IsFunction)

(operation)

Returns: nothing

The arguments are a category Cand function F. This operafunction Fthe category for the operation tion adds the given to basic Tensor Product To Internal Hom Left Adjunct Morphism With Given Internal Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f,i)\mapsto$ TensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom(a, b, f, i).

1.14.59 AddTensorProductToInternalHomLeftAdjunctionIsomorphism (for IsCap-Category, IsFunction)

- \triangleright AddTensorProductToInternalHomLeftAdjunctionIsomorphism(C, F) (operation)
- ${\tt \triangleright} \ \, \texttt{AddTensorProductToInternalHomLeftAdjunctionIsomorphism(\textit{C}, \textit{F, weight})} \quad (\textit{operation})$

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 TensorProductToInternalHomLeftAdjunctionIsomorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto TensorProductToInternalHomLeftAdjunctionIsomorphism(<math>a,b,c$).

1.14.60 AddTensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddTensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects(C, F)

(operation)

 $> AddTensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects(\textit{C, F,} weight) \\ (operation)$

Returns: nothing

F. This The arguments function operaare category Cand adds the given function Fthe category for the basic operation tion to Tensor Product To Internal Hom Left Adjunction Isomorphism With Given Objects.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ TensorProductToInternalHomLeftAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r).

1.14.61 AddTensorProductToInternalHomRightAdjunctMorphism (for IsCapCategory, IsFunction)

 ${} \hspace{0.2cm} \hspace{0.2cm$

 $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductToInternalHomRightAdjunctMorphism}(\mathit{C},\ \mathit{F},\ \mathit{weight}) \qquad \qquad (operation)$

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 TensorProductToInternalHomRightAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f)\mapsto TensorProductToInternalHomRightAdjunctMorphism(a,b,f)$.

1.14.62 AddTensorProductToInternalHomRightAdjunctMorphismWithGivenInternalHom (for IsCapCategory, IsFunction)

▷ AddTensorProductToInternalHomRightAdjunctMorphismWithGivenInternalHom(C, F)

(operation)

▷ AddTensorProductToInternalHomRightAdjunctMorphismWithGivenInternalHom(C, F,
weight)
(operation)

Returns: nothing

arguments The category Cfunction F. This operaare and tion adds given function to the category for the basic operation Tensor Product To Internal Hom Right Adjunct Morphism With Given Internal Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f,i)\mapsto$ TensorProductToInternalHomRightAdjunctMorphismWithGivenInternalHom(a, b, f, i).

1.14.63 AddTensorProductToInternalHomRightAdjunctionIsomorphism (for IsCap-Category, IsFunction)

- \triangleright AddTensorProductToInternalHomRightAdjunctionIsomorphism(C, F) (operation)
- ▷ AddTensorProductToInternalHomRightAdjunctionIsomorphism(C, F, weight) (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 TensorProductToInternalHomRightAdjunctionIsomorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto$ TensorProductToInternalHomRightAdjunctionIsomorphism(a,b,c).

1.14.64 AddTensorProductToInternalHomRightAdjunctionIsomorphismWithGivenObjects (for IsCapCategory, IsFunction)

 $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductToInternalHomRightAdjunctionIsomorphismWithGivenObjects}(\mathit{C}, \\ F, \ \mathit{weight}) \\ \qquad \qquad \qquad \qquad \qquad (\mathsf{operation}) \\$

Returns: nothing

The arguments category Cand function F. This are a operafunction the given Fto the category for the basic operation TensorProductToInternalHomRightAdjunctionIsomorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ TensorProductToInternalHomRightAdjunctionIsomorphismWithGivenObjects(s, a, b, c, r).

1.14.65 AddUniversalPropertyOfDual (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(t,a,alpha)\mapsto UniversalPropertyOfDual(t,a,alpha)$.

1.14.66 AddCoDualOnMorphisms (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \text{CoDualOnMorphisms}(alpha)$.

1.14.67 AddCoDualOnMorphismsWithGivenCoDuals (for IsCapCategory, 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 CoDualOnMorphismsWithGivenCoDuals. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,r) \mapsto \text{CoDualOnMorphismsWithGivenCoDuals}(s,alpha,r)$.

1.14.68 AddCoDualOnObjects (for IsCapCategory, 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 CoDualOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{CoDualOnObjects}(a)$.

1.14.69 AddCoDualityTensorProductCompatibilityMorphism (for IsCapCategory, IsFunction)

```
▷ AddCoDualityTensorProductCompatibilityMorphism(C, F) (operation)
▷ AddCoDualityTensorProductCompatibilityMorphism(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto CoDualityTensorProductCompatibilityMorphism(a,b)$.

1.14.70 AddCoDualityTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

Returns: nothing

The arguments category CF. This are a and function operafor function Fto the category the tion adds the given basic operation CoDualityTensorProductCompatibilityMorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ ${\tt CoDualityTensorProductCompatibilityMorphismWithGivenObjects}(s,a,b,r).$

1.14.71 AddCoLambdaElimination (for IsCapCategory, 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 CoLambdaElimination. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (a,b,alpha) \mapsto \text{CoLambdaElimination}(a,b,alpha)$.

1.14.72 AddCoLambdaIntroduction (for IsCapCategory, IsFunction)

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

▷ AddCoLambdaIntroduction(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower

1.14.73 AddCoclosedEvaluationForCoDual (for IsCapCategory, IsFunction)

weight = less complex = faster execution). $F:(alpha) \mapsto \texttt{CoLambdaIntroduction}(alpha)$.

▷ AddCoclosedEvaluationForCoDual(C, F) (operation)
▷ AddCoclosedEvaluationForCoDual(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto CoclosedEvaluationForCoDual(a)$.

1.14.74 AddCoclosedEvaluationForCoDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

```
▷ AddCoclosedEvaluationForCoDualWithGivenTensorProduct(C, F) (operation)
▷ AddCoclosedEvaluationForCoDualWithGivenTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto \texttt{CoclosedEvaluationForCoDualWithGivenTensorProduct}(s,a,r)$.

1.14.75 AddCoclosedMonoidalLeftCoevaluationMorphism (for IsCapCategory, Is-Function)

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

▷ AddCoclosedMonoidalLeftCoevaluationMorphism(C, F, weight) (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 CoclosedMonoidalLeftCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computa-

tional complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \texttt{CoclosedMonoidalLeftCoevaluationMorphism}(a,b)$.

1.14.76 AddCoclosedMonoidalLeftCoevaluationMorphismWithGivenSource (for Is-CapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddCoclosedMonoidalLeftCoevaluationMorphismWithGivenSource({\it C}, {\it F}) $ & (operation) \\ $ > $ AddCoclosedMonoidalLeftCoevaluationMorphismWithGivenSource({\it C}, {\it F}, {\it weight}) $ & (operation) \\ \end{tabular}$

Returns: nothing

The arguments Care and a function F. This operaa category tion the given function Fto the category for the basic operation ${\tt Coclosed Monoidal Left Coevaluation Morphism With Given Source}.$ Optionally, (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto$ CoclosedMonoidalLeftCoevaluationMorphismWithGivenSource(a, b, s).

1.14.77 AddCoclosedMonoidalLeftEvaluationMorphism (for IsCapCategory, IsFunction)

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

▷ AddCoclosedMonoidalLeftEvaluationMorphism(C, F, weight) (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 CoclosedMonoidalLeftEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \text{CoclosedMonoidalLeftEvaluationMorphism}(a,b)$.

1.14.78 AddCoclosedMonoidalLeftEvaluationMorphismWithGivenRange (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 CoclosedMonoidalLeftEvaluationMorphismWithGivenRange. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r)\mapsto \texttt{CoclosedMonoidalLeftEvaluationMorphismWithGivenRange}(a,b,r).$

1.14.79 AddCoclosedMonoidalRightCoevaluationMorphism (for IsCapCategory, 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 CoclosedMonoidalRightCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto CoclosedMonoidalRightCoevaluationMorphism(a,b)$.

1.14.80 AddCoclosedMonoidalRightCoevaluationMorphismWithGivenSource (for Is-CapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddCoclosedMonoidalRightCoevaluationMorphismWithGivenSource({\it C}, {\it F}) $ & (operation) \\ $ > $ AddCoclosedMonoidalRightCoevaluationMorphismWithGivenSource({\it C}, {\it F}, {\it weight}) $ & (operation) \\ \end{tabular}$

Returns: nothing

The arguments category Cand function F. This are a operation adds the given function Fto the category for the basic operation ${\tt Coclosed Monoidal Right Coevaluation Morphism With Given Source}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto$ CoclosedMonoidalRightCoevaluationMorphismWithGivenSource(a, b, s).

1.14.81 AddCoclosedMonoidalRightEvaluationMorphism (for IsCapCategory, Is-Function)

▷ AddCoclosedMonoidalRightEvaluationMorphism(C, F) (operation)
▷ AddCoclosedMonoidalRightEvaluationMorphism(C, F, weight) (operation)
□ Processing and thing

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 CoclosedMonoidalRightEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto CoclosedMonoidalRightEvaluationMorphism(a,b)$.

1.14.82 AddCoclosedMonoidalRightEvaluationMorphismWithGivenRange (for Is-CapCategory, IsFunction)

 $\begin{tabular}{l} \rhd AddCoclosedMonoidalRightEvaluationMorphismWithGivenRange(C, F) & (operation) \\ \rhd AddCoclosedMonoidalRightEvaluationMorphismWithGivenRange(C, F, $weight) & (operation) \\ \end{tabular}$

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 CoclosedMonoidalRightEvaluationMorphismWithGivenRange. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r)\mapsto CoclosedMonoidalRightEvaluationMorphismWithGivenRange(<math>a,b,r$).

1.14.83 AddInternalCoHomOnMorphisms (for IsCapCategory, IsFunction)

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 InternalCoHomOnMorphisms. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha,beta) \mapsto InternalCoHomOnMorphisms(alpha,beta)$.

1.14.84 AddInternalCoHomOnMorphismsWithGivenInternalCoHoms (for IsCap-Category, IsFunction)

▷ AddInternalCoHomOnMorphismsWithGivenInternalCoHoms(C, F) (operation)
▷ AddInternalCoHomOnMorphismsWithGivenInternalCoHoms(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). F: $(s, alpha, beta, r) \mapsto$ InternalCoHomOnMorphismsWithGivenInternalCoHoms(s, alpha, beta, r).

1.14.85 AddInternalCoHomOnObjects (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \text{InternalCoHomOnObjects}(a,b)$.

1.14.86 AddInternalCoHomTensorProductCompatibilityMorphism (for IsCapCategory, IsFunction)

▷ AddInternalCoHomTensorProductCompatibilityMorphism(C, F) (operation)
▷ AddInternalCoHomTensorProductCompatibilityMorphism(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(list) \mapsto InternalCoHomTensorProductCompatibilityMorphism(list)$.

68

(operation)

AddInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

ation)

▷ AddInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects(C, F, weight) (operation)

Returns: nothing

The arguments function F. This are category Cand operagiven function Fthe category the operation tion adds the to for basic InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(source, list, range) \mapsto$ InternalCoHomTensorProductCompatibilityMorphismWithGivenObjects(source, list, range).

AddInternalCoHomToTensorProductLeftAdjunctMorphism (for IsCapCate-1.14.88 gory, IsFunction)

▷ AddInternalCoHomToTensorProductLeftAdjunctMorphism(C, F) (operation) ▷ AddInternalCoHomToTensorProductLeftAdjunctMorphism(C, F, weight)

Returns: nothing

The arguments are a category C and a function F. This operation adds the given function Fto the category for the basic operation InternalCoHomToTensorProductLeftAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,f)\mapsto$ InternalCoHomToTensorProductLeftAdjunctMorphism(a, c, f).

1.14.89 ${\bf Add Internal Co Hom To Tensor Product Left Adjunct Morphism With Given Tensor Product}$ (for IsCapCategory, IsFunction)

▷ AddInternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(C, F) (operation)

▷ AddInternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(C, F, weight) (operation)

Returns: nothing

The arguments category Cfunction F. This operaare and given tion adds the function Fto the category for the basic operation InternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct. ally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,f,t)\mapsto$ InternalCoHomToTensorProductLeftAdjunctMorphismWithGivenTensorProduct(a, c, f, t).

1.14.90 AddInternalCoHomToTensorProductRightAdjunctMorphism (for IsCapCategory, IsFunction)

- ▷ AddInternalCoHomToTensorProductRightAdjunctMorphism(C, F) (operation)
- ▷ AddInternalCoHomToTensorProductRightAdjunctMorphism(C, F, weight) (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 InternalCoHomToTensorProductRightAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f)\mapsto$ InternalCoHomToTensorProductRightAdjunctMorphism(a,b,f).

1.14.91 AddInternalCoHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct (for IsCapCategory, IsFunction)

- ▷ AddInternalCoHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct(C, F)
 (operation)

Returns: nothing

The arguments are a category Cand function F. This opera-Fthe given function to the category for the basic operation adds Internal Co Hom To Tensor Product Right Adjunct Morphism With Given Tensor Product.tionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f,t)\mapsto$ InternalCoHomToTensorProductRightAdjunctMorphismWithGivenTensorProduct(a, b, f, t).

1.14.92 AddIsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ >$ AddIsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit(C, F) & (operation) \\ $ >$ AddIsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit(C, F, $weight) & (operation) \\ \end{tabular}$

Returns: nothing

The arguments category Cand function F. This operaare a given operation tion adds function Fto the category for the basic IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit. Optionally, 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto {\tt IsomorphismFromCoDualObjectToInternalCoHomFromTensorUnit}(a).$

1.14.93 AddIsomorphismFromInternalCoHomFromTensorUnitToCoDualObject (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ >$ AddIsomorphismFromInternalCoHomFromTensorUnitToCoDualObject({\it C}, {\it F})$ & (operation) \\ $ >$ AddIsomorphismFromInternalCoHomFromTensorUnitToCoDualObject({\it C}, {\it F}, {\it weight})$ & (operation) \\ \end{tabular}$

Returns: nothing

The arguments CF. This are category and function operagiven function Fto the category the operation the basic Isomorphism From Internal CoHom From Tensor Unit To CoDual Object.Optionally, 100) can be specified which should roughly correspond to the compuweight (default:

tational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto \mathtt{IsomorphismFromInternalCoHomFromTensorUnitToCoDualObject}(a).$

1.14.94 AddIsomorphismFromInternalCoHomToObject (for IsCapCategory, IsFunction)

 ${\hspace{-0.2cm}\triangleright\hspace{0.1cm}} \verb| AddIsomorphismFromInternalCoHomToObject({\it C, F}) \\ \\ (operation)$

▷ AddIsomorphismFromInternalCoHomToObject(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto IsomorphismFromInternalCoHomToObject(a)$.

1.14.95 AddIsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom (for IsCapCategory, IsFunction)

 ${\color{blue} \triangleright} \ \, \mathsf{AddIsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom(\mathit{C, F})} \qquad (operation)$

▷ AddIsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom(C, F, weight)

(operation)

Returns: nothing

The arguments are category Cand function F. This operation given function to the category the basic operation Isomorphism From Internal CoHom ToObject With Given Internal CoHom.Optionally, 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto \mathtt{IsomorphismFromInternalCoHomToObjectWithGivenInternalCoHom}(a,s).$

1.14.96 AddIsomorphismFromObjectToInternalCoHom (for IsCapCategory, IsFunction)

 ${\hspace{-0.2cm}\triangleright\hspace{0.1cm}} \verb| AddIsomorphismFromObjectToInternalCoHom({\it C, F}) \\ \\ (operation)$

 ${\tt \triangleright} \ {\tt AddIsomorphismFromObjectToInternalCoHom}({\tt C, F, weight})$

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto IsomorphismFromObjectToInternalCoHom(a)$.

1.14.97 AddIsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom (for IsCapCategory, IsFunction)

- \triangleright AddIsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom(C, F) (operation)
- ▷ AddIsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom(C, F, weight)

(operation)

Returns: nothing

The arguments category CF. are and function This operathe given function Fto the category for the basic operation Optionally, Isomorphism From Object To Internal CoHom With Given Internal CoHom.weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto \mathtt{IsomorphismFromObjectToInternalCoHomWithGivenInternalCoHom}(a,r).$

1.14.98 AddMonoidalPostCoComposeMorphism (for IsCapCategory, IsFunction)

▷ AddMonoidalPostCoComposeMorphism(C, F)

(operation)

▷ AddMonoidalPostCoComposeMorphism(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \texttt{MonoidalPostCoComposeMorphism}(a,b,c)$.

1.14.99 AddMonoidalPostCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMonoidalPostCoComposeMorphismWithGivenObjects(C, F)

(operation)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto MonoidalPostCoComposeMorphismWithGivenObjects(s,a,b,c,r)$.

1.14.100 AddMonoidalPreCoComposeMorphism (for IsCapCategory, IsFunction)

▷ AddMonoidalPreCoComposeMorphism(C, F)

(operation)

▷ AddMonoidalPreCoComposeMorphism(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto MonoidalPreCoComposeMorphism(a,b,c)$.

1.14.101 AddMonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

(operation)

▷ AddMonoidalPreCoComposeMorphismWithGivenObjects(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto MonoidalPreCoComposeMorphismWithGivenObjects(s,a,b,c,r)$.

1.14.102 AddMorphismFromCoBidual (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{MorphismFromCoBidual}(a)$.

1.14.103 AddMorphismFromCoBidualWithGivenCoBidual (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto \text{MorphismFromCoBidualWithGivenCoBidual}(a,s)$.

1.14.104 AddMorphismFromInternalCoHomToTensorProduct (for IsCapCategory, IsFunction)

▷ AddMorphismFromInternalCoHomToTensorProduct(C, F) (operation)

▷ AddMorphismFromInternalCoHomToTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromInternalCoHomToTensorProduct(a,b)$.

1.14.105 AddMorphismFromInternalCoHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddMorphismFromInternalCoHomToTensorProductWithGivenObjects({\it C}, {\it F}) $ & operation) \\ $ > $ AddMorphismFromInternalCoHomToTensorProductWithGivenObjects({\it C}, {\it F}, {\it weight}) $ & operation) \\ \end{tabular}$

Returns: nothing

The function arguments are a category Cand F. This operation adds the given function to the category for the basic operation ${\tt MorphismFromInternalCoHomToTensorProductWithGivenObjects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ ${\tt MorphismFromInternalCoHomToTensorProductWithGivenObjects}(s,a,b,r).$

1.14.106 AddTensorProductToInternalCoHomLeftAdjunctMorphism (for IsCapCategory, IsFunction)

- ${} \hspace*{0.2cm} \hspace$
- $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductToInternalCoHomLeftAdjunctMorphism}(\mathit{C},\ \mathit{F},\ \mathit{weight}) \qquad \qquad (operation)$

Returns: nothing

The arguments are

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation TensorProductToInternalCoHomLeftAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g)\mapsto$ TensorProductToInternalCoHomLeftAdjunctMorphism(b,c,g).

1.14.107 AddTensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom (for IsCapCategory, IsFunction)

- ▷ AddTensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom(C, F)
 (operation)
- ightharpoonup AddTensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom(C, F, weight) (operation)

Returns: nothing

arguments Cfunction F. This The category and operaare adds the given function Fto the category for the basic operation Tensor Product To Internal Co Hom Left Adjunct Morphism With Given Internal Co Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g,i)\mapsto$ TensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom(b, c, g, i).

1.14.108 AddTensorProductToInternalCoHomRightAdjunctMorphism (for IsCap-Category, IsFunction)

 ${\color{blue} \triangleright} \ \, \mathsf{AddTensorProductToInternalCoHomRightAdjunctMorphism}(\mathit{C}, \ \mathit{F}) \qquad \qquad (\mathsf{operation})$

(operation)

▷ AddTensorProductToInternalCoHomRightAdjunctMorphism(C, F, weight)
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 TensorProductToInternalCoHomRightAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g)\mapsto TensorProductToInternalCoHomRightAdjunctMorphism(b,c,g)$.

1.14.109 AddTensorProductToInternalCoHomRightAdjunctMorphismWithGivenInternalCoHom (for IsCapCategory, IsFunction)

 $\begin{tabular}{l} $ > $ AddTensorProductToInternalCoHomRightAdjunctMorphismWithGivenInternalCoHom(C, \\ F) \end{tabular}$

Returns: nothing

This The arguments Cfunction F. operaare category and adds given function Fthe category the operation tion the to for basic Tensor Product To Internal CoHomRight Adjunct Morphism With Given Internal CoHom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g,i)\mapsto$ TensorProductToInternalCoHomRightAdjunctMorphismWithGivenInternalCoHom(b, c, g, i).

1.14.110 AddUniversalPropertyOfCoDual (for IsCapCategory, IsFunction)

hd AddUniversalPropertyOfCoDual(C, F)

 ${\tt \triangleright} \ {\tt AddUniversalPropertyOfCoDual}({\tt C}, \ {\tt F}, \ {\tt weight})$

(operation) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(t,a,alpha) \mapsto UniversalPropertyOfCoDual(t,a,alpha)$.

1.14.111 AddIsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit (for IsCapCategory, IsFunction)

▷ AddIsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit(C, F) (operation)
▷ AddIsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit(C, F,
weight) (operation)

Returns: nothing

arguments Cfunction F. This are a category and a operagiven function Fto the category basic operation Isomorphism From Left Dual Object To Left Internal Hom Into Tensor Unit.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto \mathtt{IsomorphismFromLeftDualObjectToLeftInternalHomIntoTensorUnit}(a).$

1.14.112 AddIsomorphismFromLeftInternalHomIntoTensorUnitToLeftDualObject (for IsCapCategory, IsFunction)

Returns: nothing

The function arguments are a category Cand F. This operation adds the given function to the category the basic operation Isomorphism From Left Internal Hom Into Tensor Unit To Left Dual Object.Optionally, 100) can be specified which should roughly correspond to the coma weight (default: putational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto {\tt IsomorphismFromLeftInternalHomIntoTensorUnitToLeftDualObject}(a).$

1.14.113 AddIsomorphismFromLeftInternalHomToObject (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddIsomorphismFromLeftInternalHomToObject({\it C}, {\it F}) & (operation) \\ $ > $ AddIsomorphismFromLeftInternalHomToObject({\it C}, {\it F}, {\it weight}) & (operation) \\ \end{tabular}$

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 IsomorphismFromLeftInternalHomToObject. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto IsomorphismFromLeftInternalHomToObject(a)$.

1.14.114 AddIsomorphismFromLeftInternalHomToObjectWithGivenLeftInternalHom (for IsCapCategory, IsFunction)

 \triangleright AddIsomorphismFromLeftInternalHomToObjectWithGivenLeftInternalHom(C, F) (operation)

Returns: nothing

The arguments function F. This are category Cand operaa given function the the Fto category for the basic operation $Isomorphism From Left Internal Hom ToObject With {\tt GivenLeftInternal} Hom. \\$ Optionally, 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto \mathtt{IsomorphismFromLeftInternalHomToObjectWithGivenLeftInternalHom}(a,s).$

1.14.115 AddIsomorphismFromObjectToLeftInternalHom (for IsCapCategory, 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 IsomorphismFromObjectToLeftInternalHom. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto IsomorphismFromObjectToLeftInternalHom(a)$.

1.14.116 AddIsomorphismFromObjectToLeftInternalHomWithGivenLeftInternalHom (for IsCapCategory, IsFunction)

 \triangleright AddIsomorphismFromObjectToLeftInternalHomWithGivenLeftInternalHom(C, F) (operation)

Returns: nothing

F. The arguments function This are category Cand operagiven function F the category the operation tion adds the to for basic Isomorphism From Object To Left Internal Hom With Given Left Internal Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto \mathtt{IsomorphismFromObjectToLeftInternalHomWithGivenLeftInternalHom}(a,r).$

1.14.117 AddLeftClosedMonoidalCoevaluationMorphism (for IsCapCategory, IsFunction)

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 LeftClosedMonoidalCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \text{LeftClosedMonoidalCoevaluationMorphism}(a,b)$.

1.14.118 AddLeftClosedMonoidalCoevaluationMorphismWithGivenRange (for Is-CapCategory, 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 LeftClosedMonoidalCoevaluationMorphismWithGivenRange. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r)\mapsto \text{LeftClosedMonoidalCoevaluationMorphismWithGivenRange}(a,b,r).$

1.14.119 AddLeftClosedMonoidalEvaluationForLeftDual (for IsCapCategory, Is-Function)

The arguments are a category C and a function F. This operation adds the given function F to the category for the basic operation LeftClosedMonoidalEvaluationForLeftDual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computa-

tional complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{LeftClosedMonoidalEvaluationForLeftDual}(a)$.

1.14.120 AddLeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

ightharpoonup AddLeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct(C, F) (operation)

 $\qquad \qquad \triangleright \ \, \mathsf{AddLeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct}(\mathit{C},\ \mathit{F},\\ \mathit{weight}) \qquad \qquad \qquad (operation)$

Returns: nothing

The arguments CF. This are a category and function operaoperation function tion adds the given Fto the category the basic ${\tt LeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto$ LeftClosedMonoidalEvaluationForLeftDualWithGivenTensorProduct(s, a, r).

1.14.121 AddLeftClosedMonoidalEvaluationMorphism (for IsCapCategory, IsFunction)

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 LeftClosedMonoidalEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \text{LeftClosedMonoidalEvaluationMorphism}(a,b).$

1.14.122 AddLeftClosedMonoidalEvaluationMorphismWithGivenSource (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 LeftClosedMonoidalEvaluationMorphismWithGivenSource. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto \text{LeftClosedMonoidalEvaluationMorphismWithGivenSource}(a,b,s).$

1.14.123 AddLeftClosedMonoidalLambdaElimination (for IsCapCategory, IsFunction)

▷ AddLeftClosedMonoidalLambdaElimination(C, F) (operation)

▷ AddLeftClosedMonoidalLambdaElimination(C, F, weight) (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 LeftClosedMonoidalLambdaElimination. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,alpha) \mapsto \text{LeftClosedMonoidalLambdaElimination}(a,b,alpha)$.

1.14.124 AddLeftClosedMonoidalLambdaIntroduction (for IsCapCategory, IsFunction)

 $\qquad \qquad \triangleright \ \, \mathsf{AddLeftClosedMonoidalLambdaIntroduction}(\mathit{C}, \ \mathit{F}) \qquad \qquad (\mathsf{operation})$

▷ AddLeftClosedMonoidalLambdaIntroduction(C, F, weight)

(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 LeftClosedMonoidalLambdaIntroduction. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha) \mapsto LeftClosedMonoidalLambdaIntroduction(alpha)$.

1.14.125 AddLeftClosedMonoidalPostComposeMorphism (for IsCapCategory, IsFunction)

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 LeftClosedMonoidalPostComposeMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \text{LeftClosedMonoidalPostComposeMorphism}(a,b,c).$

1.14.126 AddLeftClosedMonoidalPostComposeMorphismWithGivenObjects (for Is-CapCategory, IsFunction)

 $\begin{tabular}{ll} $ \rhd$ AddLeftClosedMonoidalPostComposeMorphismWithGivenObjects(\it{C}, \it{F})$ & (operation) \\ $ \rhd$ AddLeftClosedMonoidalPostComposeMorphismWithGivenObjects(\it{C}, \it{F}, \it{weight})$ & (operation) \\ \end{tabular}$

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 LeftClosedMonoidalPostComposeMorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto \text{LeftClosedMonoidalPostComposeMorphismWithGivenObjects}(s,a,b,c,r).$

1.14.127 AddLeftClosedMonoidalPreComposeMorphism (for IsCapCategory, Is-Function)

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 LeftClosedMonoidalPreComposeMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \text{LeftClosedMonoidalPreComposeMorphism}(a,b,c).$

1.14.128 AddLeftClosedMonoidalPreComposeMorphismWithGivenObjects (for Is-CapCategory, IsFunction)

- - **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 LeftClosedMonoidalPreComposeMorphismWithGivenObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto \text{LeftClosedMonoidalPreComposeMorphismWithGivenObjects}(s,a,b,c,r).$

1.14.129 AddLeftDualOnMorphisms (for IsCapCategory, IsFunction)

▷ AddLeftDualOnMorphisms(C, F) (operation)

▷ AddLeftDualOnMorphisms(C, F, weight) (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 LeftDualOnMorphisms. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \text{LeftDualOnMorphisms}(alpha)$.

1.14.130 AddLeftDualOnMorphismsWithGivenLeftDuals (for IsCapCategory, IsFunction)

▷ AddLeftDualOnMorphismsWithGivenLeftDuals(C, F) (operation)

▷ AddLeftDualOnMorphismsWithGivenLeftDuals(C, F, weight) (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 LeftDualOnMorphismsWithGivenLeftDuals. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,r)\mapsto \text{LeftDualOnMorphismsWithGivenLeftDuals}(s,alpha,r).$

1.14.131 AddLeftDualOnObjects (for IsCapCategory, IsFunction)

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 LeftDualOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{LeftDualOnObjects}(a)$.

1.14.132 AddLeftInternalHomOnMorphisms (for IsCapCategory, IsFunction)

```
▷ AddLeftInternalHomOnMorphisms(C, F) (operation)
▷ AddLeftInternalHomOnMorphisms(C, F, weight) (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 LeftInternalHomOnMorphisms. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha,beta) \mapsto \text{LeftInternalHomOnMorphisms}(alpha,beta)$.

1.14.133 AddLeftInternalHomOnMorphismsWithGivenLeftInternalHoms (for Is-CapCategory, IsFunction)

▷ AddLeftInternalHomOnMorphismsWithGivenLeftInternalHoms(C, F) (operation)
▷ AddLeftInternalHomOnMorphismsWithGivenLeftInternalHoms(C, F, weight) (operation)
Returns: nothing

The arguments are a category Cand a function This operation adds the given function the category the basic operation $LeftInternal HomOnMorphisms With {\tt GivenLeftInternal Homs}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s, alpha, beta, r) \mapsto$ LeftInternalHomOnMorphismsWithGivenLeftInternalHoms(s, alpha, beta, r).

1.14.134 AddLeftInternalHomOnObjects (for IsCapCategory, 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 LeftInternalHomOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \text{LeftInternalHomOnObjects}(a,b)$.

1.14.135 AddLeftInternalHomToTensorProductAdjunctMorphism (for IsCapCategory, IsFunction)

```
▷ AddLeftInternalHomToTensorProductAdjunctMorphism(C, F) (operation)
▷ AddLeftInternalHomToTensorProductAdjunctMorphism(C, F, weight) (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 LeftInternalHomToTensorProductAdjunctMorphism. Op-

tionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g)\mapsto \texttt{LeftInternalHomToTensorProductAdjunctMorphism}(b,c,g).$

1.14.136 AddLeftInternalHomToTensorProductAdjunctMorphismWithGivenTensorProduct (for IsCapCategory, IsFunction)

(operation)

Returns: nothing

The arguments category Cand function F. This operaare a given function Fthe category for the operation adds the to basic Left Internal Hom To Tensor Product Adjunct Morphism With Given Tensor Product.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g,t)\mapsto$ LeftInternalHomToTensorProductAdjunctMorphismWithGivenTensorProduct(b, c, g, t).

1.14.137 AddMorphismFromTensorProductToLeftInternalHom (for IsCapCategory, IsFunction)

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

 ${\scriptsize \triangleright \ \, AddMorphismFromTensorProductToLeftInternalHom(\it{C}, \, F, \, weight) } \qquad \qquad (operation)$

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 MorphismFromTensorProductToLeftInternalHom. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromTensorProductToLeftInternalHom(a,b)$.

1.14.138 AddMorphismFromTensorProductToLeftInternalHomWithGivenObjects (for IsCapCategory, IsFunction)

 ${\tt \triangleright} \ \, {\tt AddMorphismFromTensorProductToLeftInternalHomWithGivenObjects(\it{C}, \it{F})} \qquad {\tt (operation)}$

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

(operation)

Returns: nothing

The arguments category Cand function F. This operaare the tion adds given function to the category for basic operation ${\tt MorphismFromTensorProductToLeftInternalHomWithGivenObjects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). MorphismFromTensorProductToLeftInternalHomWithGivenObjects(s, a, b, r).

1.14.139 AddMorphismToLeftBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToLeftBidual(C, F) (operation)

▷ AddMorphismToLeftBidual(C, F, weight) (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 MorphismToLeftBidual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{MorphismToLeftBidual}(a)$.

1.14.140 AddMorphismToLeftBidualWithGivenLeftBidual (for IsCapCategory, 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 MorphismToLeftBidualWithGivenLeftBidual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto MorphismToLeftBidualWithGivenLeftBidual(a,r)$.

1.14.141 AddTensorProductLeftDualityCompatibilityMorphism (for IsCapCategory, IsFunction)

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

▷ AddTensorProductLeftDualityCompatibilityMorphism(C, F, weight) (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 TensorProductLeftDualityCompatibilityMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto TensorProductLeftDualityCompatibilityMorphism(a,b)$.

1.14.142 AddTensorProductLeftDualityCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

 ${\color{blue} \triangleright} \ \, \mathsf{AddTensorProductLeftDualityCompatibilityMorphismWithGivenObjects}(\mathit{C},\ \mathit{F}) \quad \, (\mathsf{operation})$

 ${\tt \triangleright AddTensorProductLeftDualityCompatibilityMorphismWithGivenObjects(\it{C}, \it{F}, \it{weight})} \\ (operation)$

Returns: nothing

The arguments category Cand function F. This are a operagiven function Fthe tion adds to the category for basic operation Tensor Product Left Duality Compatibility Morphism With Given Objects.Optionally, weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ TensorProductLeftDualityCompatibilityMorphismWithGivenObjects(s, a, b, r).

1.14.143 AddTensorProductLeftInternalHomCompatibilityMorphism (for IsCapCategory, IsFunction)

> AddTensorProductLeftInternalHomCompatibilityMorphism(C, F) (operation)

 $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductLeftInternalHomCompatibilityMorphism}(\mathit{C},\ \mathit{F},\ \mathit{weight}) \qquad \qquad (operation)$

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 TensorProductLeftInternalHomCompatibilityMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(list) \mapsto TensorProductLeftInternalHomCompatibilityMorphism(list)$.

1.14.144 AddTensorProductLeftInternalHomCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

(operation)

ightharpoonup AddTensorProductLeftInternalHomCompatibilityMorphismWithGivenObjects(C, F, weight) (operation

Returns: nothing

The arguments CF. This are a category and function operaoperation function F the tion adds given to the category for basic Tensor Product Left Internal Hom Compatibility Morphism With Given Objects.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(source, list, range) \mapsto$ ${\tt TensorProductLeftInternalHomCompatibilityMorphismWithGivenObjects} (source, list, range).$

1.14.145 AddTensorProductToLeftInternalHomAdjunctMorphism (for IsCapCategory, IsFunction)

 $\qquad \qquad \triangleright \ \, \mathsf{AddTensorProductToLeftInternalHomAdjunctMorphism}(\mathit{C}, \ \mathit{F}) \qquad \qquad (\mathsf{operation})$

▷ AddTensorProductToLeftInternalHomAdjunctMorphism(C, F, weight) (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 TensorProductToLeftInternalHomAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f)\mapsto TensorProductToLeftInternalHomAdjunctMorphism(a,b,f)$.

1.14.146 AddTensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom (for IsCapCategory, IsFunction)

 $\begin{tabular}{l} $ > $ AddTensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom(C, \\ F) \end{tabular}$

 ${\tt \triangleright} \ \, {\tt AddTensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom}({\tt \textit{C}}, {\tt \textit{constant}}) \\$

F, weight) (operation)

Returns: nothing

The function arguments are a category Cand F. This operation adds given function to the category for the basic operation Tensor Product To Left Internal HomAdjunct Morphism With Given Left Internal Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,f,i)\mapsto$ TensorProductToLeftInternalHomAdjunctMorphismWithGivenLeftInternalHom(a, b, f, i).

1.14.147 AddUniversalPropertyOfLeftDual (for IsCapCategory, IsFunction)

▷ AddUniversalPropertyOfLeftDual(C, F)

(operation)

▷ AddUniversalPropertyOfLeftDual(C, F, weight)

(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 UniversalPropertyOfLeftDual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(t,a,alpha)\mapsto$ UniversalPropertyOfLeftDual(t,a,alpha).

1.14.148 AddIsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit (for IsCapCategory, IsFunction)

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

(operation)

 $> AddIsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit(\textit{C, F,} weight) \\ \qquad \qquad (operation)$

Returns: nothing

arguments Cfunction F. This are category and operafunction the given Fto the category the basic operation Isomorphism From Left CoDual Object To Left Internal CoHom From Tensor Unit.ally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto {\tt IsomorphismFromLeftCoDualObjectToLeftInternalCoHomFromTensorUnit}(a).$

1.14.149 AddIsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject (for IsCapCategory, IsFunction)

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

(operation)

▷ AddIsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject(C, F,
weight) (operation)

Returns: nothing

F. The arguments are a category Cand function This operathe function Fthe category operation given to the basic IsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject. Option-100) can be specified which should roughly correspond to the ally, a weight (default: computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto \texttt{IsomorphismFromLeftInternalCoHomFromTensorUnitToLeftCoDualObject}(a).$

1.14.150 AddIsomorphismFromLeftInternalCoHomToObject (for IsCapCategory, Is-Function)

 \triangleright AddIsomorphismFromLeftInternalCoHomToObject(C, F) (operation)

▷ AddIsomorphismFromLeftInternalCoHomToObject(C, F, weight)

(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 IsomorphismFromLeftInternalCoHomToObject. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto IsomorphismFromLeftInternalCoHomToObject(a)$.

1.14.151 AddIsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom (for IsCapCategory, IsFunction)

ightharpoonup AddIsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom(C, F)

(operation)

ightharpoonup AddIsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom(C, F, weight) (operation

Returns: nothing

The arguments CF. This are a category and function operafunction F the operation tion adds the given to the category for basic Isomorphism From Left Internal CoHom ToObject With Given Left Internal CoHom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). ${\tt IsomorphismFromLeftInternalCoHomToObjectWithGivenLeftInternalCoHom}(a,s).$

1.14.152 AddIsomorphismFromObjectToLeftInternalCoHom (for IsCapCategory, IsFunction)

 \triangleright AddIsomorphismFromObjectToLeftInternalCoHom(C, F) (operation)

▷ AddIsomorphismFromObjectToLeftInternalCoHom(C, F, weight) (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 IsomorphismFromObjectToLeftInternalCoHom. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto IsomorphismFromObjectToLeftInternalCoHom(a)$.

1.14.153 AddIsomorphismFromObjectToLeftInternalCoHomWithGivenLeftInternalCoHom (for IsCapCategory, IsFunction)

 $ightharpoonup AddIsomorphismFromObjectToLeftInternalCoHomWithGivenLeftInternalCoHom(\it{C}, F)$

(operation)

ightharpoonup AddIsomorphismFromObjectToLeftInternalCoHomWithGivenLeftInternalCoHom(C, F, weight) (operation)

Returns: nothing

The function arguments are a category Cand F. This operagiven tion adds the function to the category the operation Isomorphism From Object To Left Internal CoHom With Given Left Internal CoHom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). Isomorphism From Object To Left Internal CoHom With Given Left Internal CoHom (a, r).

1.14.154 AddLeftCoDualOnMorphisms (for IsCapCategory, IsFunction)

▷ AddLeftCoDualOnMorphisms(C, F) (operation)

▷ AddLeftCoDualOnMorphisms(C, F, weight) (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 LeftCoDualOnMorphisms. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \texttt{LeftCoDualOnMorphisms}(alpha)$.

1.14.155 AddLeftCoDualOnMorphismsWithGivenLeftCoDuals (for IsCapCategory, IsFunction)

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 LeftCoDualOnMorphismsWithGivenLeftCoDuals. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,r)\mapsto LeftCoDualOnMorphismsWithGivenLeftCoDuals(s,alpha,r).$

1.14.156 AddLeftCoDualOnObjects (for IsCapCategory, IsFunction)

▷ AddLeftCoDualOnObjects(C, F) (operation)

▷ AddLeftCoDualOnObjects(C, F, weight) (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 LeftCoDualOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{LeftCoDualOnObjects}(a)$.

1.14.157 AddLeftCoDualityTensorProductCompatibilityMorphism (for IsCapCategory, IsFunction)

▷ AddLeftCoDualityTensorProductCompatibilityMorphism(C, F) (operation)

▷ AddLeftCoDualityTensorProductCompatibilityMorphism(C, F, weight) (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 LeftCoDualityTensorProductCompatibilityMorphism.

Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b) \mapsto \texttt{LeftCoDualityTensorProductCompatibilityMorphism}(a,b)$.

1.14.158 AddLeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

- \triangleright AddLeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects(C, F) (operation)

Returns: nothing

The arguments category Cand function F. This are a operagiven function Fthe category the operation adds the to for basic Left CoDuality Tensor Product Compatibility Morphism With Given Objects.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ LeftCoDualityTensorProductCompatibilityMorphismWithGivenObjects(s, a, b, r).

1.14.159 AddLeftCoclosedMonoidalCoevaluationMorphism (for IsCapCategory, IsFunction)

- - **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 LeftCoclosedMonoidalCoevaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto LeftCoclosedMonoidalCoevaluationMorphism(a,b)$.

1.14.160 AddLeftCoclosedMonoidalCoevaluationMorphismWithGivenSource (for Is-CapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddLeftCoclosedMonoidalCoevaluationMorphismWithGivenSource({\it C}, {\it F}) $ & operation) \\ $ > $ AddLeftCoclosedMonoidalCoevaluationMorphismWithGivenSource({\it C}, {\it F}, {\it weight}) $ & operation) \\ \end{tabular}$

Returns: nothing

The arguments category Cand function F. This operaare tion adds given function to the category for the basic operation $Left {\tt Coclosed Monoidal Coevaluation Morphism With Given Source}.$ Optionally, (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,s)\mapsto$ LeftCoclosedMonoidalCoevaluationMorphismWithGivenSource(a, b, s).

1.14.161 AddLeftCoclosedMonoidalEvaluationForLeftCoDual (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 LeftCoclosedMonoidalEvaluationForLeftCoDual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{LeftCoclosedMonoidalEvaluationForLeftCoDual}(a)$.

1.14.162 AddLeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

 $\qquad \qquad \triangleright \ \, \mathsf{AddLeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct}(\mathit{C},\ \mathit{F})$

(operation)

▷ AddLeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct(C, F, weight)
(operation

Returns: nothing

The arguments F. This are a category Cand function operafunction the operation tion adds the given Fto the category for basic ${\tt LeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto$ LeftCoclosedMonoidalEvaluationForLeftCoDualWithGivenTensorProduct(s, a, r).

1.14.163 AddLeftCoclosedMonoidalEvaluationMorphism (for IsCapCategory, Is-Function)

ightharpoonup AddLeftCoclosedMonoidalEvaluationMorphism(C, F) (operation)

ightharpoonup AddLeftCoclosedMonoidalEvaluationMorphism(C, F, weight) (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 LeftCoclosedMonoidalEvaluationMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \text{LeftCoclosedMonoidalEvaluationMorphism}(a,b).$

1.14.164 AddLeftCoclosedMonoidalEvaluationMorphismWithGivenRange (for Is-CapCategory, IsFunction)

- ightharpoonup AddLeftCoclosedMonoidalEvaluationMorphismWithGivenRange(C, F) (operation)
- > AddLeftCoclosedMonoidalEvaluationMorphismWithGivenRange(C, F, weight) (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 LeftCoclosedMonoidalEvaluationMorphismWithGivenRange.

Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,r) \mapsto \texttt{LeftCoclosedMonoidalEvaluationMorphismWithGivenRange}(a,b,r)$.

1.14.165 AddLeftCoclosedMonoidalLambdaElimination (for IsCapCategory, IsFunction)

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

▷ AddLeftCoclosedMonoidalLambdaElimination(C, F, weight) (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 LeftCoclosedMonoidalLambdaElimination. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,alpha) \mapsto \text{LeftCoclosedMonoidalLambdaElimination}(a,b,alpha)$.

1.14.166 AddLeftCoclosedMonoidalLambdaIntroduction (for IsCapCategory, Is-Function)

```
▷ AddLeftCoclosedMonoidalLambdaIntroduction(C, F) (operation)
▷ AddLeftCoclosedMonoidalLambdaIntroduction(C, F, weight) (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 LeftCoclosedMonoidalLambdaIntroduction. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha) \mapsto LeftCoclosedMonoidalLambdaIntroduction(alpha)$.

1.14.167 AddLeftCoclosedMonoidalPostCoComposeMorphism (for IsCapCategory, IsFunction)

```
▷ AddLeftCoclosedMonoidalPostCoComposeMorphism(C, F) (operation)
▷ AddLeftCoclosedMonoidalPostCoComposeMorphism(C, F, weight) (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 LeftCoclosedMonoidalPostCoComposeMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \text{LeftCoclosedMonoidalPostCoComposeMorphism}(a,b,c).$

1.14.168 AddLeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

```
 \begin{tabular}{ll} $ > $ AddLeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects({\it C}, {\it F})$ & (operation) \\ $ > $ AddLeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects({\it C}, {\it F}, {\it weight})$ & (operation) \\ \hline \end{tabular}
```

Returns: nothing

The function This arguments are a category Cand F. operation adds given function to the category for the basic operation $Left Coclosed Monoidal Post CoCompose Morphism With {\tt Given Objects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ LeftCoclosedMonoidalPostCoComposeMorphismWithGivenObjects(s, a, b, c, r).

1.14.169 AddLeftCoclosedMonoidalPreCoComposeMorphism (for IsCapCategory, IsFunction)

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 LeftCoclosedMonoidalPreCoComposeMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \text{LeftCoclosedMonoidalPreCoComposeMorphism}(a,b,c).$

1.14.170 AddLeftCoclosedMonoidalPreCoComposeMorphismWithGivenObjects (for IsCapCategory, IsFunction)

Returns: nothing

F. arguments Cfunction This category and operaare a function the category the basic operation tion adds the given to $Left {\tt Coclosed Monoidal Pre CoCompose Morphism With Given Objects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto$ LeftCoclosedMonoidalPreCoComposeMorphismWithGivenObjects(s,a,b,c,r).

1.14.171 AddLeftInternalCoHomOnMorphisms (for IsCapCategory, IsFunction)

▷ AddLeftInternalCoHomOnMorphisms(C, F) (operation)
▷ AddLeftInternalCoHomOnMorphisms(C, F, weight) (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 LeftInternalCoHomOnMorphisms. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha,beta)\mapsto \text{LeftInternalCoHomOnMorphisms}(alpha,beta)$.

1.14.172 AddLeftInternalCoHomOnMorphismsWithGivenLeftInternalCoHoms (for IsCapCategory, IsFunction)

Returns: nothing

The arguments are category Cand function This operagiven the function the category basic operation tion adds to for the LeftInternal CoHomOnMorphisms With Given LeftInternal CoHoms.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,alpha,beta,r)\mapsto$ LeftInternalCoHomOnMorphismsWithGivenLeftInternalCoHoms(s, alpha, beta, r).

1.14.173 AddLeftInternalCoHomOnObjects (for IsCapCategory, IsFunction)

(operation)

 ${\tt \triangleright} \ {\tt AddLeftInternalCoHomOnObjects}({\tt C}, \ {\tt F}, \ {\tt weight})$

(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 LeftInternalCoHomOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \texttt{LeftInternalCoHomOnObjects}(a,b)$.

1.14.174 AddLeftInternalCoHomTensorProductCompatibilityMorphism (for IsCap-Category, IsFunction)

- ${} \hspace*{0.2cm} \hspace$
- ${\tt \triangleright} \ \, {\tt AddLeftInternalCoHomTensorProductCompatibilityMorphism(\textit{C}, \textit{F, weight})} \quad ({\tt operation})$

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 LeftInternalCoHomTensorProductCompatibilityMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(list) \mapsto \text{LeftInternalCoHomTensorProductCompatibilityMorphism}(list)$.

1.14.175 AddLeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddLeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects(C, F)

(operation)

 $\label{eq:compatibility} $$ $$ AddLeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects({\it C, F, weight})$$ (operation) $$$

Returns: nothing

The arguments are category Cand function This operaa given function Fto the category for the LeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects. Optionally, a

weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(source, list, range) \mapsto \texttt{LeftInternalCoHomTensorProductCompatibilityMorphismWithGivenObjects}(source, list, range).$

1.14.176 AddLeftInternalCoHomToTensorProductAdjunctMorphism (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddLeftInternalCoHomToTensorProductAdjunctMorphism(\it{C}, \it{F}) & operation \\ $ > $ AddLeftInternalCoHomToTensorProductAdjunctMorphism(\it{C}, \it{F}, weight) \\ \end{tabular} $ & operation \\ \end{tabular}$

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 LeftInternalCoHomToTensorProductAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,f)\mapsto \text{LeftInternalCoHomToTensorProductAdjunctMorphism}(a,c,f).$

1.14.177 AddLeftInternalCoHomToTensorProductAdjunctMorphismWithGivenTensorProduct (for IsCapCategory, IsFunction)

- $\label{eq:continuous} $$ $ $$ Add Left Internal CoHom To Tensor Product Adjunct Morphism With Given Tensor Product (C, F) $$ (operation) $$$

Returns: nothing

F. The arguments are category Cand function This operaa given adds the function Fto the category for the basic operation Left Internal CoHomToTensor Product Adjunct Morphism With Given Tensor Product.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,c,f,t)\mapsto$ LeftInternalCoHomToTensorProductAdjunctMorphismWithGivenTensorProduct(a, c, f, t).

1.14.178 AddMorphismFromLeftCoBidual (for IsCapCategory, IsFunction)

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 MorphismFromLeftCoBidual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{MorphismFromLeftCoBidual}(a)$.

1.14.179 AddMorphismFromLeftCoBidualWithGivenLeftCoBidual (for IsCapCategory, IsFunction)

- ${} \hspace*{0.2cm} \hspace$
- ightharpoonup AddMorphismFromLeftCoBidualWithGivenLeftCoBidual(C, F, weight)

(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 MorphismFromLeftCoBidualWithGivenLeftCoBidual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto MorphismFromLeftCoBidualWithGivenLeftCoBidual(a,s)$.

1.14.180 AddMorphismFromLeftInternalCoHomToTensorProduct (for IsCapCategory, IsFunction)

- \triangleright AddMorphismFromLeftInternalCoHomToTensorProduct(C, F) (operation)
- ${\tt \triangleright} \ \, {\tt AddMorphismFromLeftInternalCoHomToTensorProduct(\textit{C}, \textit{F, weight)}} \qquad \qquad ({\tt operation})$

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 MorphismFromLeftInternalCoHomToTensorProduct. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromLeftInternalCoHomToTensorProduct(a,b)$.

1.14.181 AddMorphismFromLeftInternalCoHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

▷ AddMorphismFromLeftInternalCoHomToTensorProductWithGivenObjects(C, F) (operation)
▷ AddMorphismFromLeftInternalCoHomToTensorProductWithGivenObjects(C, F, weight) (operation)

Returns: nothing

F. The arguments are a category Cand a function This operation function Fto the category for the basic operation adds the given ${\tt MorphismFromLeftInternalCoHomToTensorProductWithGivenObjects}.$ Optionally, weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ MorphismFromLeftInternalCoHomToTensorProductWithGivenObjects(s, a, b, r).

1.14.182 AddTensorProductToLeftInternalCoHomAdjunctMorphism (for IsCapCategory, IsFunction)

- \triangleright AddTensorProductToLeftInternalCoHomAdjunctMorphism(C, F) (operation)
- ${\tt \hspace*{-0.5cm} \hspace*{-0.$

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 TensorProductToLeftInternalCoHomAdjunctMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g)\mapsto TensorProductToLeftInternalCoHomAdjunctMorphism(b,c,g)$.

1.14.183 AddTensorProductToLeftInternalCoHomAdjunctMorphismWithGivenLeftInternalCoHom (for IsCapCategory, IsFunction)

▷ AddTensorProductToLeftInternalCoHomAdjunctMorphismWithGivenLeftInternalCoHom(C, F)
(operation)

Returns: nothing

The arguments function F. This are category Cand operafunction Fthe the operation tion adds the given to category for basic Tensor Product To Left Internal CoHom Adjunct Morphism With Given Left Internal CoHom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(b,c,g,i)\mapsto$ TensorProductToLeftInternalCoHomAdjunctMorphismWithGivenLeftInternalCoHom(b, c, g, i).

1.14.184 AddUniversalPropertyOfLeftCoDual (for IsCapCategory, IsFunction)

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 UniversalPropertyOfLeftCoDual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(t,a,alpha)\mapsto$ UniversalPropertyOfLeftCoDual(t,a,alpha).

1.14.185 AddAssociatorLeftToRight (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \texttt{AssociatorLeftToRight}(a,b,c)$.

1.14.186 AddAssociatorLeftToRightWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddAssociatorLeftToRightWithGivenTensorProducts(C, F) (operation)

▷ AddAssociatorLeftToRightWithGivenTensorProducts(C, F, weight) (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 AssociatorLeftToRightWithGivenTensorProducts. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto AssociatorLeftToRightWithGivenTensorProducts(s,a,b,c,r)$.

1.14.187 AddAssociatorRightToLeft (for IsCapCategory, IsFunction)

▷ AddAssociatorRightToLeft(C, F) (operation)

▷ AddAssociatorRightToLeft(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b,c)\mapsto \text{AssociatorRightToLeft}(a,b,c)$.

1.14.188 AddAssociatorRightToLeftWithGivenTensorProducts (for IsCapCategory, IsFunction)

▷ AddAssociatorRightToLeftWithGivenTensorProducts(C, F) (operation)

▷ AddAssociatorRightToLeftWithGivenTensorProducts(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,c,r)\mapsto AssociatorRightToLeftWithGivenTensorProducts(s,a,b,c,r)$.

1.14.189 AddLeftUnitor (for IsCapCategory, 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 LeftUnitor. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{LeftUnitor}(a)$.

1.14.190 AddLeftUnitorInverse (for IsCapCategory, IsFunction)

```
▷ AddLeftUnitorInverse(C, F) (operation)
▷ AddLeftUnitorInverse(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{LeftUnitorInverse}(a)$.

1.14.191 AddLeftUnitorInverseWithGivenTensorProduct (for IsCapCategory, Is-Function)

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

▷ AddLeftUnitorInverseWithGivenTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto \text{LeftUnitorInverseWithGivenTensorProduct}(a,r).$

1.14.192 AddLeftUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

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

▷ AddLeftUnitorWithGivenTensorProduct(C, F, weight) (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 LeftUnitorWithGivenTensorProduct. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto \text{LeftUnitorWithGivenTensorProduct}(a,s)$.

1.14.193 AddRightUnitor (for IsCapCategory, 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 RightUnitor. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{RightUnitor}(a)$.

1.14.194 AddRightUnitorInverse (for IsCapCategory, IsFunction)

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

▷ AddRightUnitorInverse(C, F, weight) (operation)

■ Potential and the properties of the
```

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{RightUnitorInverse}(a)$.

1.14.195 AddRightUnitorInverseWithGivenTensorProduct (for IsCapCategory, Is-Function)

```
▷ AddRightUnitorInverseWithGivenTensorProduct(C, F) (operation)
▷ AddRightUnitorInverseWithGivenTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto RightUnitorInverseWithGivenTensorProduct(a,r)$.

1.14.196 AddRightUnitorWithGivenTensorProduct (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto RightUnitorWithGivenTensorProduct(a,s)$.

1.14.197 AddTensorProductOnMorphisms (for IsCapCategory, IsFunction)

AddTensorProductOnMorphisms(C, F) (operation)

 AddTensorProductOnMorphisms(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(alpha,beta) \mapsto TensorProductOnMorphisms(alpha,beta)$.

1.14.198 AddTensorProductOnMorphismsWithGivenTensorProducts (for IsCapCategory, 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 TensorProductOnMorphismsWithGivenTensorProducts. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). F: $(s, alpha, beta, r) \mapsto \text{TensorProductOnMorphismsWithGivenTensorProducts}(s, alpha, beta, r)$.

1.14.199 AddTensorProductOnObjects (for IsCapCategory, 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 TensorProductOnObjects. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (arg2, arg3) \mapsto \text{TensorProductOnObjects}(arg2, arg3)$.

1.14.200 AddTensorUnit (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: () \mapsto \texttt{TensorUnit}()$.

1.14.201 AddCoevaluationForDual (for IsCapCategory, IsFunction)

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

▷ AddCoevaluationForDual(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto \texttt{CoevaluationForDual}(a)$.

1.14.202 AddCoevaluationForDualWithGivenTensorProduct (for IsCapCategory, IsFunction)

```
▷ AddCoevaluationForDualWithGivenTensorProduct(C, F) (operation)
▷ AddCoevaluationForDualWithGivenTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto \text{CoevaluationForDualWithGivenTensorProduct}(s,a,r)$.

1.14.203 AddIsomorphismFromInternalHomToTensorProductWithDualObject (for IsCapCategory, IsFunction)

 \triangleright AddIsomorphismFromInternalHomToTensorProductWithDualObject(C, F) (operation) \triangleright AddIsomorphismFromInternalHomToTensorProductWithDualObject(C, F, weight) (operation)

Returns: nothing

The arguments are category Cand function F. This operaa function Fthe category for basic operation tion adds the given to the Isomorphism From Internal Hom To Tensor Product With Dual Object.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto$ IsomorphismFromInternalHomToTensorProductWithDualObject(a,b).

1.14.204 AddIsomorphismFromTensorProductWithDualObjectToInternalHom (for IsCapCategory, IsFunction)

Returns: nothing

The arguments are category Cand function This operagiven operation adds the function the category for the basic tion F to Isomorphism From Tensor Product With Dual Object To Internal Hom.Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto$ IsomorphismFromTensorProductWithDualObjectToInternalHom(a,b).

1.14.205 AddMorphismFromBidual (for IsCapCategory, 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 MorphismFromBidual. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{MorphismFromBidual}(a)$.

1.14.206 AddMorphismFromBidualWithGivenBidual (for IsCapCategory, IsFunction)

▷ AddMorphismFromBidualWithGivenBidual(C, F) (operation)
 ▷ AddMorphismFromBidualWithGivenBidual(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,s)\mapsto MorphismFromBidualWithGivenBidual(a,s)$.

1.14.207 AddMorphismFromInternalHomToTensorProduct (for IsCapCategory, IsFunction)

▷ AddMorphismFromInternalHomToTensorProduct(C, F) (operation)
▷ AddMorphismFromInternalHomToTensorProduct(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromInternalHomToTensorProduct(a,b)$.

1.14.208 AddMorphismFromInternalHomToTensorProductWithGivenObjects (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddMorphismFromInternalHomToTensorProductWithGivenObjects(\it{C}, \it{F}) $ & (operation) \\ > $ AddMorphismFromInternalHomToTensorProductWithGivenObjects(\it{C}, \it{F}, \it{weight}) $ & (operation) \\ \end{tabular}$

Returns: nothing

The arguments are category Cand function This operathe given function the category for the basic operation tion adds to ${\tt MorphismFromInternalHomToTensorProductWithGivenObjects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r)\mapsto$ MorphismFromInternalHomToTensorProductWithGivenObjects(s, a, b, r).

1.14.209 AddRankMorphism (for IsCapCategory, 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 RankMorphism. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{RankMorphism}(a)$.

1.14.210 AddTensorProductInternalHomCompatibilityMorphismInverse (for IsCap-Category, IsFunction)

▷ AddTensorProductInternalHomCompatibilityMorphismInverse(C, F) (operation)
▷ AddTensorProductInternalHomCompatibilityMorphismInverse(C, F, weight) (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 TensorProductInternalHomCompatibilityMorphismInverse. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(list) \mapsto TensorProductInternalHomCompatibilityMorphismInverse(list)$.

1.14.211 AddTensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects (for IsCapCategory, IsFunction)

- - **Returns:** nothing

The arguments are category Cand function This operaa given function Fto the category for the operation TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects. Optionally,

a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (source, list, range) \mapsto TensorProductInternalHomCompatibilityMorphismInverseWithGivenObjects(source, list, range).$

1.14.212 AddTraceMap (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \text{TraceMap}(alpha)$.

1.14.213 AddCoRankMorphism (for IsCapCategory, IsFunction)

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

▷ AddCoRankMorphism(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \text{CoRankMorphism}(a)$.

1.14.214 AddCoTraceMap (for IsCapCategory, IsFunction)

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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F: (alpha) \mapsto \text{CoTraceMap}(alpha)$.

1.14.215 AddCoclosedCoevaluationForCoDual (for IsCapCategory, IsFunction)

```
▷ AddCoclosedCoevaluationForCoDual(C, F) (operation)
▷ AddCoclosedCoevaluationForCoDual(C, F, weight) (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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a)\mapsto CoclosedCoevaluationForCoDual(a)$.

1.14.216 AddCoclosedCoevaluationForCoDualWithGivenTensorProduct (for IsCap-Category, IsFunction)

 ${\color{blue} \triangleright} \ \, \mathsf{AddCoclosedCoevaluationForCoDualWithGivenTensorProduct}\left(\mathit{C},\ \mathit{F}\right) \qquad \qquad (\mathsf{operation})$

 $\qquad \qquad \triangleright \ \, \mathsf{AddCoclosedCoevaluationForCoDualWithGivenTensorProduct}(\mathit{C, F, weight}) \quad (\mathsf{operation})$

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 CoclosedCoevaluationForCoDualWithGivenTensorProduct. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(s,a,r)\mapsto CoclosedCoevaluationForCoDualWithGivenTensorProduct(s,a,r)$.

1.14.217 AddInternalCoHomTensorProductCompatibilityMorphismInverse (for Is-CapCategory, IsFunction)

Returns: nothing

This The arguments F. category Cand function operaare the given function F the category the operation adds to for basic ${\tt InternalCoHomTensorProductCompatibilityMorphismInverse}.$ Optionally, (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $Internal CoHom Tensor Product Compatibility Morphism Inverse (\it list).$

1.14.218 AddInternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects (for IsCapCategory, IsFunction)

▶ AddInternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects(C, F)
(operation)

Returns: nothing

The arguments are category Cand function F. This operagiven adds function to the category the operation Internal Co Hom Tensor Product Compatibility Morphism Inverse With Given Objects.(default: Optionally, weight 100) can specified which be should roughly correspond to the of the function (lower computational complexity execution). F: $(source, list, range) \mapsto$ weight less complex faster InternalCoHomTensorProductCompatibilityMorphismInverseWithGivenObjects(source, list, range).

1.14.219 AddIsomorphismFromInternalCoHomToTensorProductWithCoDualObject (for IsCapCategory, IsFunction)

- ▷ AddIsomorphismFromInternalCoHomToTensorProductWithCoDualObject(C, F, weight)

(operation)

Returns: nothing

The arguments F. are category Cand function This operaadds the given function Fto the category for the basic operation Isomorphism From Internal CoHom To Tensor Product With CoDual Object.Optionally, weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \texttt{IsomorphismFromInternalCoHomToTensorProductWithCoDualObject}(a,b).$

1.14.220 AddIsomorphismFromTensorProductWithCoDualObjectToInternalCoHom (for IsCapCategory, IsFunction)

ightharpoonup AddIsomorphismFromTensorProductWithCoDualObjectToInternalCoHom(C, F) (operation) ightharpoonup AddIsomorphismFromTensorProductWithCoDualObjectToInternalCoHom(C, F, weight)

(operation)

Returns: nothing

The arguments category Cand function F. This are operafunction tion adds given to the category for the basic operation Isomorphism From Tensor Product With CoDual Object To Internal CoHom.Optionally, 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto \mathtt{IsomorphismFromTensorProductWithCoDualObjectToInternalCoHom}(a,b).$

1.14.221 AddMorphismFromTensorProductToInternalCoHom (for IsCapCategory, IsFunction)

 ${\hspace{0.2cm}} \hspace{0.2cm} \hspace{0.2cm}$

▷ AddMorphismFromTensorProductToInternalCoHom(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,b)\mapsto MorphismFromTensorProductToInternalCoHom(a,b)$.

1.14.222 AddMorphismFromTensorProductToInternalCoHomWithGivenObjects (for IsCapCategory, IsFunction)

 $\begin{tabular}{ll} $ > $ AddMorphismFromTensorProductToInternalCoHomWithGivenObjects({\it C}, {\it F}) $ & (operation) \\ $ > $ AddMorphismFromTensorProductToInternalCoHomWithGivenObjects({\it C}, {\it F}, {\it weight}) $ & (operation) \\ \end{tabular}$

Returns: nothing

The arguments Cfunction F. This are category and operagiven function Fto the category the basic operation the ${\tt MorphismFromTensorProductToInternalCoHomWithGivenObjects}.$ Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational com-

plexity of the function (lower weight = less complex = faster execution). $F:(s,a,b,r) \mapsto \texttt{MorphismFromTensorProductToInternalCoHomWithGivenObjects}(s,a,b,r).$

1.14.223 AddMorphismToCoBidual (for IsCapCategory, IsFunction)

 $hd \ \ Add Morphism To Co Bidual({\it C, F})$

(operation)

 ${\scriptstyle \rhd} \ \, {\tt AddMorphismToCoBidual}(\mathit{C, F, weight}) \\$

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a) \mapsto \texttt{MorphismToCoBidual}(a)$.

1.14.224 AddMorphismToCoBidualWithGivenCoBidual (for IsCapCategory, IsFunction)

▷ AddMorphismToCoBidualWithGivenCoBidual(C, F)

(operation)

▷ AddMorphismToCoBidualWithGivenCoBidual(C, F, weight)

(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. Optionally, a weight (default: 100) can be specified which should roughly correspond to the computational complexity of the function (lower weight = less complex = faster execution). $F:(a,r)\mapsto \text{MorphismToCoBidualWithGivenCoBidual}(a,r)$.

Chapter 2

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

106

- 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

```
▷ 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 $\alpha: a \rightarrow b$
- 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 ClosedMonoidalCategoriesTestWithGiven

```
\triangleright ClosedMonoidalCategoriesTestWithGiven(cat, a, b, c, d, alpha, beta) (function)
```

The arguments are

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

107

• a morphism $\beta: c \to d$

This function checks for some *WithGiven operations declared in ClosedMonoidalCategories.gd if they are 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 CoclosedMonoidalCategoriesTest

```
▷ 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 ε : coHom $(a,b) \to 1$
- a morphism ζ : coHom $(c,d) \to 1$

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.

2.1.6 CoclosedMonoidalCategoriesTestWithGiven

```
\triangleright CoclosedMonoidalCategoriesTestWithGiven(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 $\beta: c \to d$

This function checks for some *WithGiven operations declared in CoclosedMonoidalCategories.gd if they are 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 LeftClosedMonoidalCategoriesTest

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 LeftClosedMonoidalCategories.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 LeftClosedMonoidalCategoriesTestWithGiven

```
▷ LeftClosedMonoidalCategoriesTestWithGiven(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 $\beta: c \to d$

This function checks for some *WithGiven operationS declared in LeftClosedMonoidalCategories.gd if they are 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.

2.1.9 LeftCoclosedMonoidalCategoriesTest

```
\triangleright LeftCoclosedMonoidalCategoriesTest(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 $\gamma: 1 \rightarrow a \otimes b$
- a morphism $\delta: 1 \rightarrow c \otimes d$

```
• a morphism \varepsilon : coHom(a,b) \to 1
```

```
• a morphism \zeta : coHom(c,d) \rightarrow 1
```

This function checks for every operation declared in LeftCoclosedMonoidalCategories.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.

${\bf 2.1.10} \quad Left Coclosed Monoidal Categories Test With Given$

```
▷ LeftCoclosedMonoidalCategoriesTestWithGiven(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 $\beta: c \to d$

This function checks for some *WithGiven operations declared in LeftCoclosedMonoidalCategories.gd if they are 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.

${\bf 2.1.11} \quad Monoidal Categories Tensor Product On Objects And Tensor Unit Test$

▷ MonoidalCategoriesTensorProductOnObjectsAndTensorUnitTest(cat, a, b) (function)

The arguments are

- a CAP category cat
- objects a, b

111

This function checks for every operation declared in MonoidalCategoriesTensorProductOnObject-sAndTensorUnit.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.

2.1.12 MonoidalCategoriesTest

```
\triangleright 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 $\beta: c \to d$

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.

2.1.13 RigidSymmetricClosedMonoidalCategoriesTest

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

The arguments are

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

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.

2.1.14 RigidSymmetricCoclosedMonoidalCategoriesTest

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

The arguments are

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

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.

Chapter 3

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 <code>key_val_rec</code> to create a new closed monoidal structure. It generates the necessary files in the package <code>package_name</code> using the file-correspondence table <code>files_rec</code>. See the implementation for details.

3.2.2 WriteFileForLeftClosedMonoidalStructure

Returns: nothing

This functions uses the dictionary key_val_rec to create a new left 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

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_name using the file-correspondence table files_rec. See the implementation for details.

3.3.2 WriteFileForLeftCoclosedMonoidalStructure

 $\label{lem:condition} $$ $$ $$ WriteFileForLeftCoclosedMonoidalStructure(key_val_rec, package_name, files_rec) $$ $$ $$ (function) $$$

Returns: nothing

This functions uses the dictionary key_val_rec to create a new left 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.

Chapter 4

The terminal category with multiple objects

This is an example of a category which is created using ${\tt Category Constructor}$ out of no input.

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

4.2 GAP Categories

Chapter 5

Legacy Operations and Synonyms

5.1 Legacy operations

5.1.1 CoclosedCoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

(operation)

This is a legacy operation for CoclosedMonoidalLeftCoevaluationMorphism(b, a), i.e., with the first and second argument interchanged.

5.1.2 CoclosedCoevaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedCoevaluationMorphismWithGivenSource(a, b, s) (operation)

This is a legacy operation for CoclosedMonoidalLeftCoevaluationMorphismWithGivenSource(b, a, s), i.e., with the first and second argument interchanged.

5.1.3 CoclosedEvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoclosedEvaluationMorphism(a, b) (operation)

This is a legacy operation for CoclosedMonoidalLeftEvaluationMorphism(b, a), i.e., with the first and second argument interchanged.

5.1.4 CoclosedEvaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

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

This is a legacy operation for CoclosedMonoidalLeftEvaluationMorphismWithGivenRange(b, a, r), i.e., with the first and second argument interchanged.

5.1.5 CoevaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ CoevaluationMorphism(a, b)

(operation)

This is a legacy operation for $ClosedMonoidalLeftCoevaluationMorphism(\ b,\ a\),\ i.e.,\ with$ the first and second argument interchanged.

5.1.6 CoevaluationMorphismWithGivenRange (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ CoevaluationMorphismWithGivenRange(a, b, r)

(operation)

This is a legacy operation for ClosedMonoidalLeftCoevaluationMorphismWithGivenRange(b, a, r), i.e., with the first and second argument interchanged.

5.2 Synonyms for legacy operations

5.2.1 EvaluationMorphism (for IsCapCategoryObject, IsCapCategoryObject)

▷ EvaluationMorphism(arg1, arg2)

(operation)

This is a synonym for ClosedMonoidalLeftEvaluationMorphism.

5.2.2 EvaluationMorphismWithGivenSource (for IsCapCategoryObject, IsCapCategoryObject, IsCapCategoryObject)

▷ EvaluationMorphismWithGivenSource(arg1, arg2, arg3)

(operation)

This is a synonym for ClosedMonoidalLeftEvaluationMorphismWithGivenSource.

5.2.3 InternalCoHomToTensorProductAdjunctionMap (for IsObject)

 $\quad \, \triangleright \,\, \texttt{InternalCoHomToTensorProductAdjunctionMap}(arg)$

(operation)

This is a synonym for InternalCoHomToTensorProductLeftAdjunctMorphism.

5.2.4 InternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsObject)

▷ InternalCoHomToTensorProductAdjunctionMapWithGivenTensorProduct(arg) (operation)

 $This is a synonym for {\tt InternalCoHomToTensorProductLeftAdjunctionMapWithGivenTensorProduct}.$

5.2.5 InternalHomToTensorProductAdjunctionMap (for IsObject)

▷ InternalHomToTensorProductAdjunctionMap(arg)

(operation)

This is a synonym for InternalHomToTensorProductLeftAdjunctMorphism.

5.2.6 InternalHomToTensorProductAdjunctionMapWithGivenTensorProduct (for IsObject)

□ InternalHomToTensorProductAdjunctionMapWithGivenTensorProduct(arg)

(operation)

 $This is a synonym for {\tt Internal HomToTensor Product LeftAdjunction Map With Given Tensor Product}.$

5.2.7 TensorProductToInternalCoHomAdjunctionMap (for IsObject)

(operation)

 $This is a synonym for {\tt TensorProductToInternalCoHomLeftAdjunctMorphism}.$

5.2.8 TensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom (for IsObject)

□ TensorProductToInternalCoHomAdjunctionMapWithGivenInternalCoHom(arg) (operation)

This is a synonym for TensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom.

5.2.9 TensorProductToInternalHomAdjunctionMap (for IsObject)

□ TensorProductToInternalHomAdjunctionMap(arg)

(operation)

 $This is a synonym for {\tt TensorProductToInternalHomLeftAdjunctMorphism}.$

5.2.10 TensorProductToInternalHomAdjunctionMapWithGivenInternalHom (for IsObject)

→ TensorProductToInternalHomAdjunctionMapWithGivenInternalHom(arg)

(operation)

 $This is a synonym for {\tt TensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom.}\\$

5.2.11 InternalCoHomToTensorProductLeftAdjunctionMap (for IsObject)

▷ InternalCoHomToTensorProductLeftAdjunctionMap(arg)

(operation)

This is a synonym for InternalCoHomToTensorProductLeftAdjunctMorphism.

5.2.12 InternalHomToTensorProductLeftAdjunctionMap (for IsObject)

▷ InternalHomToTensorProductLeftAdjunctionMap(arg)

(operation)

This is a synonym for InternalHomToTensorProductLeftAdjunctMorphism.

5.2.13 TensorProductToInternalCoHomLeftAdjunctionMap (for IsObject)

→ TensorProductToInternalCoHomLeftAdjunctionMap(arg)

(operation)

This is a synonym for TensorProductToInternalCoHomLeftAdjunctMorphism.

5.2.14 TensorProductToInternalCoHomLeftAdjunctionMapWithGivenInternalCoHom (for IsObject)

 $This is a synonym for {\tt TensorProductToInternalCoHomLeftAdjunctMorphismWithGivenInternalCoHom.}\\$

5.2.15 TensorProductToInternalHomLeftAdjunctionMap (for IsObject)

→ TensorProductToInternalHomLeftAdjunctionMap(arg)

(operation)

This is a synonym for TensorProductToInternalHomLeftAdjunctMorphism.

5.2.16 TensorProductToInternalHomLeftAdjunctionMapWithGivenInternalHom (for IsObject)

 ${\tt \begin{tabular}{l} $ \vdash$ Tensor Product To Internal HomLeft Adjunction Map With Given Internal Hom (arg) & (operation) & (op$

 $This is a synonym for {\tt TensorProductToInternalHomLeftAdjunctMorphismWithGivenInternalHom.}\\$

Chapter 6

MonoidalCategories automatic generated documentation

6.1 MonoidalCategories automatic generated documentation of properties

6.1.1 IsBraidedMonoidalCategory (for IsCapCategory)

▷ IsBraidedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being braided monoidal.

6.1.2 IsClosedMonoidalCategory (for IsCapCategory)

▷ IsClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being (bi)closed monoidal.

6.1.3 IsCoclosedMonoidalCategory (for IsCapCategory)

▷ IsCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being (bi)coclosed monoidal.

6.1.4 IsLeftClosedMonoidalCategory (for IsCapCategory)

▷ IsLeftClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being left closed monoidal.

6.1.5 IsLeftCoclosedMonoidalCategory (for IsCapCategory)

▷ IsLeftCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being coclosed monoidal.

6.1.6 IsMonoidalCategory (for IsCapCategory)

 \triangleright IsMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being monoidal.

6.1.7 IsStrictMonoidalCategory (for IsCapCategory)

▷ IsStrictMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being strict monoidal.

6.1.8 IsRigidSymmetricClosedMonoidalCategory (for IsCapCategory)

▷ IsRigidSymmetricClosedMonoidalCategory(C)

(property)

Returns: true or false

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

6.1.9 IsRigidSymmetricCoclosedMonoidalCategory (for IsCapCategory)

▷ IsRigidSymmetricCoclosedMonoidalCategory(C)

(property)

Returns: true or false

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

6.1.10 IsSymmetricClosedMonoidalCategory (for IsCapCategory)

▷ IsSymmetricClosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric closed monoidal.

6.1.11 IsSymmetricCoclosedMonoidalCategory (for IsCapCategory)

▷ IsSymmetricCoclosedMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric coclosed monoidal.

6.1.12 IsSymmetricMonoidalCategory (for IsCapCategory)

▷ IsSymmetricMonoidalCategory(C)

(property)

Returns: true or false

The property of the category C being symmetric monoidal.

Index

AddAssociatorLeftToRight	for IsCapCategory, IsFunction, IsInt, 49
for IsCapCategory, IsFunction, 94	AddClosedMonoidalLeftEvaluation-
for IsCapCategory, IsFunction, IsInt, 94	MorphismWithGivenSource
AddAssociatorLeftToRightWithGiven-	for IsCapCategory, IsFunction, 49
TensorProducts	for IsCapCategory, IsFunction, IsInt, 49
for IsCapCategory, IsFunction, 94	AddClosedMonoidalRightCoevaluation-
for IsCapCategory, IsFunction, IsInt, 94	Morphism
AddAssociatorRightToLeft	for IsCapCategory, IsFunction, 49
for IsCapCategory, IsFunction, 95	for IsCapCategory, IsFunction, IsInt, 49
for IsCapCategory, IsFunction, IsInt, 95	AddClosedMonoidalRightCoevaluation-
AddAssociatorRightToLeftWithGiven-	MorphismWithGivenRange
TensorProducts	for IsCapCategory, IsFunction, 50
for IsCapCategory, IsFunction, 95	for IsCapCategory, IsFunction, IsInt, 50
for IsCapCategory, IsFunction, IsInt, 95	AddClosedMonoidalRightEvaluation-
AddBraiding	Morphism
for IsCapCategory, IsFunction, 47	for IsCapCategory, IsFunction, 50
for IsCapCategory, IsFunction, IsInt, 47	for IsCapCategory, IsFunction, IsInt, 50
${\tt AddBraidingInverse}$	${\tt AddClosedMonoidalRightEvaluation-}$
for IsCapCategory, IsFunction, 48	MorphismWithGivenSource
for IsCapCategory, IsFunction, IsInt, 48	for IsCapCategory, IsFunction, 50
${\tt AddBraidingInverseWithGivenTensor-}$	for IsCapCategory, IsFunction, IsInt, 50
Products	${\tt AddCoclosedCoevaluationForCoDual}$
for IsCapCategory, IsFunction, 48	for IsCapCategory, IsFunction, 101
for IsCapCategory, IsFunction, IsInt, 48	for IsCapCategory, IsFunction, IsInt, 101
${\tt AddBraidingWithGivenTensorProducts}$	${\tt AddCoclosedCoevaluationForCoDualWith-}$
for IsCapCategory, IsFunction, 48	${\tt GivenTensorProduct}$
for IsCapCategory, IsFunction, IsInt, 48	for IsCapCategory, IsFunction, 102
${\tt AddClosedMonoidalLeftCoevaluation-}$	for IsCapCategory, IsFunction, IsInt, 102
Morphism	${\tt AddCoclosedEvaluationForCoDual}$
for IsCapCategory, IsFunction, 48	for IsCapCategory, IsFunction, 64
for IsCapCategory, IsFunction, IsInt, 48	for IsCapCategory, IsFunction, IsInt, 64
${\tt AddClosedMonoidalLeftCoevaluation-}$	${\tt AddCoclosedEvaluationForCoDualWith-}$
${\tt MorphismWithGivenRange}$	${\tt GivenTensorProduct}$
for IsCapCategory, IsFunction, 49	for IsCapCategory, IsFunction, 64
for IsCapCategory, IsFunction, IsInt, 49	for IsCapCategory, IsFunction, IsInt, 64
${\tt AddClosedMonoidalLeftEvaluation-}$	${\tt AddCoclosedMonoidalLeftCoevaluation-}$
Morphism	Morphism
for IsCapCategory, IsFunction, 49	for IsCapCategory, IsFunction, 64

for IsCapCategory, IsFunction, IsInt, 64	for IsCapCategory, IsFunction, IsInt, 63
AddCoclosedMonoidalLeftCoevaluation-	AddCoevaluationForDual
MorphismWithGivenSource	for IsCapCategory, IsFunction, 98
for IsCapCategory, IsFunction, 65	for IsCapCategory, IsFunction, IsInt, 98
for IsCapCategory, IsFunction, IsInt, 65	AddCoevaluationForDualWithGivenTensor-
AddCoclosedMonoidalLeftEvaluation-	Product
Morphism	for IsCapCategory, IsFunction, 98
for IsCapCategory, IsFunction, 65	for IsCapCategory, IsFunction, IsInt, 98
for IsCapCategory, IsFunction, IsInt, 65	AddCoLambdaElimination
AddCoclosedMonoidalLeftEvaluation-	for IsCapCategory, IsFunction, 63
${\tt MorphismWithGivenRange}$	for IsCapCategory, IsFunction, IsInt, 63
for IsCapCategory, IsFunction, 65	AddCoLambdaIntroduction
for IsCapCategory, IsFunction, IsInt, 65	for IsCapCategory, IsFunction, 64
AddCoclosedMonoidalRightCoevaluation-	for IsCapCategory, IsFunction, IsInt, 64
Morphism	AddCoRankMorphism
for IsCapCategory, IsFunction, 65	for IsCapCategory, IsFunction, 101
for IsCapCategory, IsFunction, IsInt, 65	for IsCapCategory, IsFunction, IsInt, 101
AddCoclosedMonoidalRightCoevaluation-	AddCoTraceMap
MorphismWithGivenSource	for IsCapCategory, IsFunction, 101
for IsCapCategory, IsFunction, 66	for IsCapCategory, IsFunction, IsInt, 101
for IsCapCategory, IsFunction, IsInt, 66	AddDualOnMorphisms
AddCoclosedMonoidalRightEvaluation-	for IsCapCategory, IsFunction, 50
Morphism	for IsCapCategory, IsFunction, IsInt, 50
for IsCapCategory, IsFunction, 66	AddDualOnMorphismsWithGivenDuals
for IsCapCategory, IsFunction, IsInt, 66	for IsCapCategory, IsFunction, 51
AddCoclosedMonoidalRightEvaluation-	for IsCapCategory, IsFunction, IsInt, 51
MorphismWithGivenRange	AddDualOnObjects
for IsCapCategory, IsFunction, 66	for IsCapCategory, IsFunction, 51
for IsCapCategory, IsFunction, IsInt, 66	for IsCapCategory, IsFunction, IsInt, 51
AddCoDualityTensorProduct-	AddEvaluationForDual
CompatibilityMorphism	for IsCapCategory, IsFunction, 51
for IsCapCategory, IsFunction, 63	for IsCapCategory, IsFunction, IsInt, 51
for IsCapCategory, IsFunction, IsInt, 63	AddEvaluationForDualWithGivenTensor-
AddCoDualityTensorProduct-	Product
CompatibilityMorphismWith-	for IsCapCategory, IsFunction, 51
GivenObjects	for IsCapCategory, IsFunction, IsInt, 51
for IsCapCategory, IsFunction, 63	AddInternalCoHomOnMorphisms
for IsCapCategory, IsFunction, IsInt, 63	for IsCapCategory, IsFunction, 67
AddCoDualOnMorphisms	for IsCapCategory, IsFunction, IsInt, 67
for IsCapCategory, IsFunction, 62	AddInternalCoHomOnMorphismsWithGiven-
for IsCapCategory, IsFunction, IsInt, 62	InternalCoHoms
AddCoDualOnMorphismsWithGivenCoDuals	for IsCapCategory, IsFunction, 67
for IsCapCategory, IsFunction, 62	for IsCapCategory, IsFunction, IsInt, 67
for IsCapCategory, IsFunction, IsInt, 62	AddInternalCoHomOnObjects
AddCoDualOnObjects	for IsCapCategory, IsFunction, 67
for IsCapCategory, IsFunction, 63	for IsCapCategory, IsFunction, IsInt, 67

AddInternalCoHomTensorProduct-	${\tt AddInternalHomToTensorProductLeft-}$
${ t Compatibility Morphism}$	${\tt AdjunctionIsomorphism}$
for IsCapCategory, IsFunction, 67	for IsCapCategory, IsFunction, 53
for IsCapCategory, IsFunction, IsInt, 67	for IsCapCategory, IsFunction, IsInt, 53
AddInternalCoHomTensorProduct-	AddInternalHomToTensorProductLeft-
${\tt CompatibilityMorphismInverse}$	AdjunctionIsomorphismWithGiven-
for IsCapCategory, IsFunction, 102	Objects
for IsCapCategory, IsFunction, IsInt, 102	for IsCapCategory, IsFunction, 53
AddInternalCoHomTensorProduct-	for IsCapCategory, IsFunction, IsInt, 53
CompatibilityMorphismInverse-	AddInternalHomToTensorProductLeft-
WithGivenObjects	AdjunctMorphism
for IsCapCategory, IsFunction, 102	for IsCapCategory, IsFunction, 52
for IsCapCategory, IsFunction, IsInt, 102	for IsCapCategory, IsFunction, IsInt, 52
AddInternalCoHomTensorProduct-	AddInternalHomToTensorProductLeft-
CompatibilityMorphismWith-	AdjunctMorphismWithGivenTensor
GivenObjects	Product
for IsCapCategory, IsFunction, 68	for IsCapCategory, IsFunction, 52
for IsCapCategory, IsFunction, IsInt, 68	for IsCapCategory, IsFunction, IsInt, 53
AddInternalCoHomToTensorProductLeft-	AddInternalHomToTensorProductRight-
AdjunctMorphism	AdjunctionIsomorphism
for IsCapCategory, IsFunction, 68	for IsCapCategory, IsFunction, 54
for IsCapCategory, IsFunction, IsInt, 68	for IsCapCategory, IsFunction, IsInt, 54
AddInternalCoHomToTensorProductLeft-	AddInternalHomToTensorProductRight-
AdjunctMorphismWithGivenTensor-	AdjunctionIsomorphismWithGiven
Product	Objects
for IsCapCategory, IsFunction, 68	for IsCapCategory, IsFunction, 54
for IsCapCategory, IsFunction, IsInt, 68	for IsCapCategory, IsFunction, IsInt, 54
AddInternalCoHomToTensorProductRight-	AddInternalHomToTensorProductRight-
AdjunctMorphism	AdjunctMorphism
for IsCapCategory, IsFunction, 68	for IsCapCategory, IsFunction, 53
for IsCapCategory, IsFunction, IsInt, 68	for IsCapCategory, IsFunction, IsInt, 53
AddInternalCoHomToTensorProductRight-	AddInternalHomToTensorProductRight-
${\tt AdjunctMorphismWithGivenTensor-}$	${\tt AdjunctMorphismWithGivenTensor}$
Product	Product
for IsCapCategory, IsFunction, 69	for IsCapCategory, IsFunction, 54
for IsCapCategory, IsFunction, IsInt, 69	for IsCapCategory, IsFunction, IsInt, 54
AddInternalHomOnMorphisms	AddIsomorphismFromCoDualObjectTo-
for IsCapCategory, IsFunction, 51	InternalCoHomFromTensorUnit
for IsCapCategory, IsFunction, IsInt, 51	for IsCapCategory, IsFunction, 69
AddInternalHomOnMorphismsWithGiven-	for IsCapCategory, IsFunction, IsInt, 69
InternalHoms	AddIsomorphismFromDualObjectTo-
for IsCapCategory, IsFunction, 52	InternalHomIntoTensorUnit
for IsCapCategory, IsFunction, IsInt, 52	for IsCapCategory, IsFunction, 54
AddInternalHomOnObjects	for IsCapCategory, IsFunction, IsInt, 54
for IsCapCategory, IsFunction, 52	AddIsomorphismFromInternalCoHomFrom-
for IsCapCategory, IsFunction, IsInt, 52	TensorUnitToCoDualObject

for IsCapCategory, IsFunction, 69	AddIsomorphismFromLeftInternalCoHom-
for IsCapCategory, IsFunction, IsInt, 69	${\tt ToObjectWithGivenLeftInternal-}$
AddIsomorphismFromInternalCoHomTo-	CoHom
Object	for IsCapCategory, IsFunction, 85
for IsCapCategory, IsFunction, 70	for IsCapCategory, IsFunction, IsInt, 85
for IsCapCategory, IsFunction, IsInt, 70	AddIsomorphismFromLeftInternalHomInto
AddIsomorphismFromInternalCoHomTo-	${ t Tensor Unit To Left Dual Object}$
${\tt ObjectWithGivenInternalCoHom}$	for IsCapCategory, IsFunction, 74
for IsCapCategory, IsFunction, 70	for IsCapCategory, IsFunction, IsInt, 74
for IsCapCategory, IsFunction, IsInt, 70	AddIsomorphismFromLeftInternalHomTo-
AddIsomorphismFromInternalCoHomTo-	Object
TensorProductWithCoDualObject	for IsCapCategory, IsFunction, 75
for IsCapCategory, IsFunction, 102	for IsCapCategory, IsFunction, IsInt, 75
for IsCapCategory, IsFunction, IsInt, 102	AddIsomorphismFromLeftInternalHomTo-
AddIsomorphismFromInternalHomInto-	$\hbox{\tt ObjectWithGivenLeftInternalHom}$
TensorUnitToDualObject	for IsCapCategory, IsFunction, 75
for IsCapCategory, IsFunction, 55	for IsCapCategory, IsFunction, IsInt, 75
for IsCapCategory, IsFunction, IsInt, 55	AddIsomorphismFromObjectToInternal-
AddIsomorphismFromInternalHomToObject	CoHom
for IsCapCategory, IsFunction, 55	for IsCapCategory, IsFunction, 70
for IsCapCategory, IsFunction, IsInt, 55	for IsCapCategory, IsFunction, IsInt, 70
AddIsomorphismFromInternalHomToObject-	AddIsomorphismFromObjectToInternalCo-
WithGivenInternalHom	HomWithGivenInternalCoHom
for IsCapCategory, IsFunction, 55	for IsCapCategory, IsFunction, 70
for IsCapCategory, IsFunction, IsInt, 55	for IsCapCategory, IsFunction, IsInt, 70
AddIsomorphismFromInternalHomToTensor-	AddIsomorphismFromObjectToInternalHom
ProductWithDualObject	for IsCapCategory, IsFunction, 56
for IsCapCategory, IsFunction, 98	for IsCapCategory, IsFunction, IsInt, 56
for IsCapCategory, IsFunction, IsInt, 98	AddIsomorphismFromObjectToInternalHom
AddIsomorphismFromLeftCoDualObjectTo-	WithGivenInternalHom
t LeftInternalCoHomFromTensorUnit	for IsCapCategory, IsFunction, 56
for IsCapCategory, IsFunction, 84	for IsCapCategory, IsFunction, IsInt, 56
for IsCapCategory, IsFunction, IsInt, 84	AddIsomorphismFromObjectToLeft-
AddIsomorphismFromLeftDualObjectTo-	InternalCoHom
$ t LeftInternal ext{HomIntoTensorUnit}$	for IsCapCategory, IsFunction, 85
for IsCapCategory, IsFunction, 74	for IsCapCategory, IsFunction, IsInt, 85
for IsCapCategory, IsFunction, IsInt, 74	AddIsomorphismFromObjectToLeft-
AddIsomorphismFromLeftInternalCoHom-	InternalCoHomWithGivenLeft-
FromTensorUnitToLeftCoDual-	InternalCoHom
Object	for IsCapCategory, IsFunction, 85
for IsCapCategory, IsFunction, 84	for IsCapCategory, IsFunction, IsInt, 85
for IsCapCategory, IsFunction, IsInt, 84	AddIsomorphismFromObjectToLeft-
AddIsomorphismFromLeftInternalCoHomTo-	InternalHom
Object	for IsCapCategory, IsFunction, 75
for IsCapCategory, IsFunction, 85	for IsCapCategory, IsFunction, IsInt, 75
for IsCapCategory, IsFunction, IsInt, 85	AddIsomorphismFromObjectToLeft-
,	···

InternalHomWithGivenLeft-	AddLeftClosedMonoidalLambda-
InternalHom	Introduction
for IsCapCategory, IsFunction, 76	for IsCapCategory, IsFunction, 78
for IsCapCategory, IsFunction, IsInt, 76	for IsCapCategory, IsFunction, IsInt, 78
AddIsomorphismFromTensorProductWithCo-	AddLeftClosedMonoidalPostCompose-
DualObjectToInternalCoHom	Morphism
for IsCapCategory, IsFunction, 103	for IsCapCategory, IsFunction, 78
for IsCapCategory, IsFunction, IsInt, 103	for IsCapCategory, IsFunction, IsInt, 78
AddIsomorphismFromTensorProductWith-	AddLeftClosedMonoidalPostCompose-
DualObjectToInternalHom	MorphismWithGivenObjects
for IsCapCategory, IsFunction, 99	for IsCapCategory, IsFunction, 78
for IsCapCategory, IsFunction, IsInt, 99	for IsCapCategory, IsFunction, IsInt, 78
AdditiveMonoidalCategoriesTest, 105	AddLeftClosedMonoidalPreCompose-
AddLambdaElimination	Morphism
for IsCapCategory, IsFunction, 56	for IsCapCategory, IsFunction, 78
for IsCapCategory, IsFunction, IsInt, 56	for IsCapCategory, IsFunction, IsInt, 78
AddLambdaIntroduction	AddLeftClosedMonoidalPreCompose-
for IsCapCategory, IsFunction, 56	MorphismWithGivenObjects
for IsCapCategory, IsFunction, IsInt, 56	for IsCapCategory, IsFunction, 79
AddLeftClosedMonoidalCoevaluation-	for IsCapCategory, IsFunction, IsInt, 79
Morphism	AddLeftCoclosedMonoidalCoevaluation-
for IsCapCategory, IsFunction, 76	Morphism
for IsCapCategory, IsFunction, IsInt, 76	for IsCapCategory, IsFunction, 87
AddLeftClosedMonoidalCoevaluation-	for IsCapCategory, IsFunction, IsInt, 87
${\tt MorphismWithGivenRange}$	AddLeftCoclosedMonoidalCoevaluation-
for IsCapCategory, IsFunction, 76	MorphismWithGivenSource
for IsCapCategory, IsFunction, IsInt, 76	for IsCapCategory, IsFunction, 87
AddLeftClosedMonoidalEvaluationFor-	for IsCapCategory, IsFunction, IsInt, 87
LeftDual	${\tt AddLeftCoclosedMonoidalEvaluationFor}$
for IsCapCategory, IsFunction, 76	LeftCoDual
for IsCapCategory, IsFunction, IsInt, 76	for IsCapCategory, IsFunction, 88
AddLeftClosedMonoidalEvaluationFor-	for IsCapCategory, IsFunction, IsInt, 88
${\tt LeftDualWithGivenTensorProduct}$	AddLeftCoclosedMonoidalEvaluation-
for IsCapCategory, IsFunction, 77	${\tt ForLeftCoDualWithGivenTensor-}$
for IsCapCategory, IsFunction, IsInt, 77	Product
AddLeftClosedMonoidalEvaluation-	for IsCapCategory, IsFunction, 88
Morphism	for IsCapCategory, IsFunction, IsInt, 88
for IsCapCategory, IsFunction, 77	${\tt AddLeftCoclosedMonoidalEvaluation-}$
for IsCapCategory, IsFunction, IsInt, 77	Morphism
AddLeftClosedMonoidalEvaluation-	for IsCapCategory, IsFunction, 88
MorphismWithGivenSource	for IsCapCategory, IsFunction, IsInt, 88
for IsCapCategory, IsFunction, 77	AddLeftCoclosedMonoidalEvaluation-
for IsCapCategory, IsFunction, IsInt, 77	${\tt MorphismWithGivenRange}$
${\tt AddLeftClosedMonoidalLambdaElimination}$	for IsCapCategory, IsFunction, 88
for IsCapCategory, IsFunction, 77	for IsCapCategory, IsFunction, IsInt, 88
for IsCapCategory, IsFunction, IsInt, 77	AddLeftCoclosedMonoidalLambda-

Elimination	GivenObjects
for IsCapCategory, IsFunction, 89	for IsCapCategory, IsFunction, 46
for IsCapCategory, IsFunction, IsInt, 89	for IsCapCategory, IsFunction, IsInt, 46
${\tt AddLeftCoclosedMonoidalLambda-}$	${\tt AddLeftDistributivityFactoring}$
Introduction	for IsCapCategory, IsFunction, 46
for IsCapCategory, IsFunction, 89	for IsCapCategory, IsFunction, IsInt, 46
for IsCapCategory, IsFunction, IsInt, 89	${\tt AddLeftDistributivityFactoringWith-}$
${\tt AddLeftCoclosedMonoidalPostCoCompose-}$	GivenObjects
Morphism	for IsCapCategory, IsFunction, 46
for IsCapCategory, IsFunction, 89	for IsCapCategory, IsFunction, IsInt, 46
for IsCapCategory, IsFunction, IsInt, 89	AddLeftDualOnMorphisms
AddLeftCoclosedMonoidalPostCoCompose-	for IsCapCategory, IsFunction, 79
MorphismWithGivenObjects	for IsCapCategory, IsFunction, IsInt, 79
for IsCapCategory, IsFunction, 89	AddLeftDualOnMorphismsWithGivenLeft-
for IsCapCategory, IsFunction, IsInt, 89	Duals
AddLeftCoclosedMonoidalPreCoCompose-	for IsCapCategory, IsFunction, 79
Morphism	for IsCapCategory, IsFunction, IsInt, 79
for IsCapCategory, IsFunction, 90	AddLeftDualOnObjects
for IsCapCategory, IsFunction, IsInt, 90	for IsCapCategory, IsFunction, 79
AddLeftCoclosedMonoidalPreCoCompose-	for IsCapCategory, IsFunction, IsInt, 79
MorphismWithGivenObjects	AddLeftInternalCoHomOnMorphisms
for IsCapCategory, IsFunction, 90	for IsCapCategory, IsFunction, 90
for IsCapCategory, IsFunction, IsInt, 90	for IsCapCategory, IsFunction, IsInt, 90
AddLeftCoDualityTensorProduct-	AddLeftInternalCoHomOnMorphismsWith-
CompatibilityMorphism	$\hbox{\tt GivenLeftInternalCoHoms}$
for IsCapCategory, IsFunction, 86	for IsCapCategory, IsFunction, 91
for IsCapCategory, IsFunction, IsInt, 86	for IsCapCategory, IsFunction, IsInt, 91
AddLeftCoDualityTensorProduct-	AddLeftInternalCoHomOnObjects
CompatibilityMorphismWith-	for IsCapCategory, IsFunction, 91
GivenObjects	for IsCapCategory, IsFunction, IsInt, 91
for IsCapCategory, IsFunction, 87	AddLeftInternalCoHomTensorProduct-
for IsCapCategory, IsFunction, IsInt, 87	CompatibilityMorphism
AddLeftCoDualOnMorphisms	for IsCapCategory, IsFunction, 91
for IsCapCategory, IsFunction, 86	for IsCapCategory, IsFunction, IsInt, 91
for IsCapCategory, IsFunction, IsInt, 86	AddLeftInternalCoHomTensorProduct-
AddLeftCoDualOnMorphismsWithGivenLeft-	CompatibilityMorphismWithGiven
CoDuals	Objects
for IsCapCategory, IsFunction, 86	for IsCapCategory, IsFunction, 91
for IsCapCategory, IsFunction, IsInt, 86	for IsCapCategory, IsFunction, IsInt, 91
AddLeftCoDualOnObjects	AddLeftInternalCoHomToTensorProduct-
for IsCapCategory, IsFunction, 86	AdjunctMorphism
for IsCapCategory, IsFunction, 86	for IsCapCategory, IsFunction, 92
AddLeftDistributivityExpanding	for IsCapCategory, IsFunction, IsInt, 92 AddLeftInternalCoHomToTensorProduct-
for IsCapCategory, IsFunction, 45	
for IsCapCategory, IsFunction, IsInt, 45	AdjunctMorphismWithGivenTensor
AddLeftDistributivityExpandingWith-	Product

for IsCapCategory, IsFunction, 92	for IsCapCategory, IsFunction, 57
for IsCapCategory, IsFunction, IsInt, 92	for IsCapCategory, IsFunction, IsInt, 57
AddLeftInternalHomOnMorphisms	AddMonoidalPreCoComposeMorphism
for IsCapCategory, IsFunction, 80	for IsCapCategory, IsFunction, 71
for IsCapCategory, IsFunction, IsInt, 80	for IsCapCategory, IsFunction, IsInt, 71
AddLeftInternalHomOnMorphismsWith-	AddMonoidalPreCoComposeMorphismWith-
GivenLeftInternalHoms	GivenObjects
for IsCapCategory, IsFunction, 80	for IsCapCategory, IsFunction, 71
for IsCapCategory, IsFunction, IsInt, 80	for IsCapCategory, IsFunction, IsInt, 71
AddLeftInternalHomOnObjects	AddMonoidalPreComposeMorphism
for IsCapCategory, IsFunction, 80	for IsCapCategory, IsFunction, 57
for IsCapCategory, IsFunction, IsInt, 80	for IsCapCategory, IsFunction, IsInt, 57
AddLeftInternalHomToTensorProduct-	AddMonoidalPreComposeMorphismWith-
AdjunctMorphism	GivenObjects
for IsCapCategory, IsFunction, 80	for IsCapCategory, IsFunction, 57
for IsCapCategory, IsFunction, IsInt, 80	for IsCapCategory, IsFunction, IsInt, 57
AddLeftInternalHomToTensorProduct-	AddMorphismFromBidual
AdjunctMorphismWithGivenTensor-	for IsCapCategory, IsFunction, 99
Product	for IsCapCategory, IsFunction, IsInt, 99
for IsCapCategory, IsFunction, 81	
, , ,	AddMorphismFromBidualWithGivenBidual
for IsCapCategory, IsFunction, IsInt, 81	for IsCapCategory, IsFunction, 99
AddLeftUnitor	for IsCapCategory, IsFunction, IsInt, 99
for IsCapCategory, IsFunction, 95	AddMorphismFromCoBidual
for IsCapCategory, IsFunction, IsInt, 95	for IsCapCategory, IsFunction, 72
AddLeftUnitorInverse	for IsCapCategory, IsFunction, IsInt, 72
for IsCapCategory, IsFunction, 95	AddMorphismFromCoBidualWithGivenCo-
for IsCapCategory, IsFunction, IsInt, 95	Bidual
AddLeftUnitorInverseWithGivenTensor-	for IsCapCategory, IsFunction, 72
Product	for IsCapCategory, IsFunction, IsInt, 72
for IsCapCategory, IsFunction, 95	${\tt AddMorphismFromInternalCoHomToTensor}$
for IsCapCategory, IsFunction, IsInt, 95	Product
AddLeftUnitorWithGivenTensorProduct	for IsCapCategory, IsFunction, 72
for IsCapCategory, IsFunction, 96	for IsCapCategory, IsFunction, IsInt, 72
for IsCapCategory, IsFunction, IsInt, 96	${\tt AddMorphismFromInternalCoHomToTensor}$
${\tt AddMonoidalPostCoComposeMorphism}$	${\tt ProductWithGivenObjects}$
for IsCapCategory, IsFunction, 71	for IsCapCategory, IsFunction, 72
for IsCapCategory, IsFunction, IsInt, 71	for IsCapCategory, IsFunction, IsInt, 72
${\tt AddMonoidalPostCoComposeMorphismWith-}$	${\tt AddMorphismFromInternalHomToTensor-}$
GivenObjects	Product
for IsCapCategory, IsFunction, 71	for IsCapCategory, IsFunction, 99
for IsCapCategory, IsFunction, IsInt, 71	for IsCapCategory, IsFunction, IsInt, 99
AddMonoidalPostComposeMorphism	${\tt AddMorphismFromInternalHomToTensor-}$
for IsCapCategory, IsFunction, 57	ProductWithGivenObjects
for IsCapCategory, IsFunction, IsInt, 57	for IsCapCategory, IsFunction, 100
AddMonoidalPostComposeMorphismWith-	for IsCapCategory, IsFunction, IsInt, 100
GivenObjects	AddMorphismFromLeftCoBidual

for IsCapCategory, IsFunction, 92	for IsCapCategory, IsFunction, IsInt, 104
for IsCapCategory, IsFunction, IsInt, 92	${\tt AddMorphismToCoBidualWithGivenCoBidual}$
AddMorphismFromLeftCoBidualWithGiven-	for IsCapCategory, IsFunction, 104
LeftCoBidual	for IsCapCategory, IsFunction, IsInt, 104
for IsCapCategory, IsFunction, 92	AddMorphismToLeftBidual
for IsCapCategory, IsFunction, IsInt, 92	for IsCapCategory, IsFunction, 82
AddMorphismFromLeftInternalCoHomTo-	for IsCapCategory, IsFunction, IsInt, 82
TensorProduct	AddMorphismToLeftBidualWithGivenLeft-
for IsCapCategory, IsFunction, 93	Bidual
for IsCapCategory, IsFunction, IsInt, 93	for IsCapCategory, IsFunction, 82
AddMorphismFromLeftInternalCoHomTo-	for IsCapCategory, IsFunction, IsInt, 82
TensorProductWithGivenObjects	AddRankMorphism
for IsCapCategory, IsFunction, 93	for IsCapCategory, IsFunction, 100
for IsCapCategory, IsFunction, IsInt, 93	for IsCapCategory, IsFunction, IsInt, 100
AddMorphismFromTensorProductTo-	AddRightDistributivityExpanding
InternalCoHom	for IsCapCategory, IsFunction, 46
for IsCapCategory, IsFunction, 103	for IsCapCategory, IsFunction, IsInt, 46
for IsCapCategory, IsFunction, IsInt, 103	AddRightDistributivityExpandingWith-
AddMorphismFromTensorProductTo-	GivenObjects
InternalCoHomWithGivenObjects	for IsCapCategory, IsFunction, 47
for IsCapCategory, IsFunction, 103	for IsCapCategory, IsFunction, IsInt, 47
for IsCapCategory, IsFunction, IsInt, 103	AddRightDistributivityFactoring
AddMorphismFromTensorProductTo-	for IsCapCategory, IsFunction, 47
InternalHom	for IsCapCategory, IsFunction, IsInt, 47
for IsCapCategory, IsFunction, 58	AddRightDistributivityFactoringWith-
for IsCapCategory, IsFunction, IsInt, 58	GivenObjects
AddMorphismFromTensorProductTo-	for IsCapCategory, IsFunction, 47
InternalHomWithGivenObjects	for IsCapCategory, IsFunction, IsInt, 47
for IsCapCategory, IsFunction, 58	AddRightUnitor
for IsCapCategory, IsFunction, IsInt, 58	for IsCapCategory, IsFunction, 96
AddMorphismFromTensorProductToLeft-	for IsCapCategory, IsFunction, IsInt, 96
InternalHom	AddRightUnitorInverse
for IsCapCategory, IsFunction, 81	for IsCapCategory, IsFunction, 96
for IsCapCategory, IsFunction, IsInt, 81	for IsCapCategory, IsFunction, IsInt, 96
AddMorphismFromTensorProductToLeft-	AddRightUnitorInverseWithGivenTensor-
InternalHomWithGivenObjects	Product
for IsCapCategory, IsFunction, 81	for IsCapCategory, IsFunction, 96
for IsCapCategory, IsFunction, IsInt, 81	for IsCapCategory, IsFunction, IsInt, 96
AddMorphismToBidual	AddRightUnitorWithGivenTensorProduct
for IsCapCategory, IsFunction, 58	for IsCapCategory, IsFunction, 97
for IsCapCategory, IsFunction, IsInt, 58	for IsCapCategory, IsFunction, IsInt, 97
AddMorphismToBidualWithGivenBidual	AddTensorProductDualityCompatibility-
for IsCapCategory, IsFunction, 58	Morphism
for IsCapCategory, IsFunction, IsInt, 58	for IsCapCategory, IsFunction, 59
AddMorphismToCoBidual	for IsCapCategory, IsFunction, IsInt, 59
	* * *
for IsCapCategory, IsFunction, 104	AddTensorProductDualityCompatibility-

${\tt MorphismWithGivenObjects}$	AddTensorProductOnObjects
for IsCapCategory, IsFunction, 59	for IsCapCategory, IsFunction, 97
for IsCapCategory, IsFunction, IsInt, 59	for IsCapCategory, IsFunction, IsInt, 97
AddTensorProductInternalHom-	${\tt AddTensorProductToInternalCoHomLeft-}$
${\tt CompatibilityMorphism}$	AdjunctMorphism
for IsCapCategory, IsFunction, 59	for IsCapCategory, IsFunction, 73
for IsCapCategory, IsFunction, IsInt, 59	for IsCapCategory, IsFunction, IsInt, 73
AddTensorProductInternalHom-	AddTensorProductToInternalCoHomLeft-
${\tt CompatibilityMorphismInverse}$	AdjunctMorphismWithGiven-
for IsCapCategory, IsFunction, 100	InternalCoHom
for IsCapCategory, IsFunction, IsInt, 100	for IsCapCategory, IsFunction, 73
AddTensorProductInternalHom-	for IsCapCategory, IsFunction, IsInt, 73
CompatibilityMorphismInverse-	AddTensorProductToInternalCoHomRight-
WithGivenObjects	AdjunctMorphism
for IsCapCategory, IsFunction, 100	for IsCapCategory, IsFunction, 73
for IsCapCategory, IsFunction, IsInt, 100	for IsCapCategory, IsFunction, IsInt, 73
AddTensorProductInternalHom-	AddTensorProductToInternalCoHomRight-
CompatibilityMorphismWith-	AdjunctMorphismWithGiven-
GivenObjects	InternalCoHom
for IsCapCategory, IsFunction, 59	for IsCapCategory, IsFunction, 74
for IsCapCategory, IsFunction, IsInt, 59	for IsCapCategory, IsFunction, IsInt, 74
AddTensorProductLeftDuality-	AddTensorProductToInternalHomLeft-
CompatibilityMorphism	AdjunctionIsomorphism
for IsCapCategory, IsFunction, 82	for IsCapCategory, IsFunction, 60
for IsCapCategory, IsFunction, IsInt, 82	for IsCapCategory, IsFunction, IsInt, 60
AddTensorProductLeftDuality-	AddTensorProductToInternalHomLeft-
CompatibilityMorphismWith-	AdjunctionIsomorphismWithGiven-
GivenObjects	Objects
for IsCapCategory, IsFunction, 82	for IsCapCategory, IsFunction, 61
for IsCapCategory, IsFunction, IsInt, 82	for IsCapCategory, IsFunction, IsInt, 61
AddTensorProductLeftInternalHom-	${\tt AddTensorProductToInternalHomLeft-}$
${\tt CompatibilityMorphism}$	AdjunctMorphism
for IsCapCategory, IsFunction, 83	for IsCapCategory, IsFunction, 60
for IsCapCategory, IsFunction, IsInt, 83	for IsCapCategory, IsFunction, IsInt, 60
AddTensorProductLeftInternalHom-	AddTensorProductToInternalHomLeft-
CompatibilityMorphismWith-	AdjunctMorphismWithGiven-
GivenObjects	${\tt InternalHom}$
for IsCapCategory, IsFunction, 83	for IsCapCategory, IsFunction, 60
for IsCapCategory, IsFunction, IsInt, 83	for IsCapCategory, IsFunction, IsInt, 60
AddTensorProductOnMorphisms	${\tt AddTensorProductToInternalHomRight-}$
for IsCapCategory, IsFunction, 97	${\tt AdjunctionIsomorphism}$
for IsCapCategory, IsFunction, IsInt, 97	for IsCapCategory, IsFunction, 61
${\tt AddTensorProductOnMorphismsWithGiven-}$	for IsCapCategory, IsFunction, IsInt, 61
TensorProducts	${\tt AddTensorProductToInternalHomRight-}$
for IsCapCategory, IsFunction, 97	${\tt AdjunctionIsomorphismWithGiven-}$
for IsCapCategory, IsFunction, IsInt, 97	Objects

for IsCapCategory, IsFunction, 62	for IsCapCategory, IsFunction, IsInt, 84
for IsCapCategory, IsFunction, IsInt, 62	AssociatorLeftToRight
AddTensorProductToInternalHomRight-	for IsCapCategoryObject, IsCapCategory-
AdjunctMorphism	Object, IsCapCategoryObject, 4
for IsCapCategory, IsFunction, 61	AssociatorLeftToRightWithGivenTensor-
for IsCapCategory, IsFunction, IsInt, 61	Products
AddTensorProductToInternalHomRight-	for IsCapCategoryObject, IsCapCategory-
AdjunctMorphismWithGiven-	Object, IsCapCategoryObject, IsCap-
InternalHom	CategoryObject, IsCapCategoryObject,
for IsCapCategory, IsFunction, 61	4
for IsCapCategory, IsFunction, IsInt, 61	AssociatorRightToLeft
AddTensorProductToLeftInternalCoHom-	for IsCapCategoryObject, IsCapCategory-
AdjunctMorphism	Object, IsCapCategoryObject, 4
for IsCapCategory, IsFunction, 93	AssociatorRightToLeftWithGivenTensor-
for IsCapCategory, IsFunction, IsInt, 93	Products
AddTensorProductToLeftInternalCoHom-	for IsCapCategoryObject, IsCapCategory-
AdjunctMorphismWithGivenLeft-	Object, IsCapCategoryObject, IsCap-
InternalCoHom	CategoryObject, IsCapCategoryObject,
for IsCapCategory, IsFunction, 94	4
for IsCapCategory, IsFunction, IsInt, 94	•
AddTensorProductToLeftInternalHom-	BraidedMonoidalCategoriesTest, 105
AdjunctMorphism	Braiding
for IsCapCategory, IsFunction, 83	for IsCapCategoryObject, IsCapCategory-
for IsCapCategory, IsFunction, IsInt, 83	Object, 7
AddTensorProductToLeftInternalHom-	BraidingInverse
AdjunctMorphismWithGivenLeft-	for IsCapCategoryObject, IsCapCategory-
InternalHom	Object, 8
for IsCapCategory, IsFunction, 83	BraidingInverseWithGivenTensorProducts
for IsCapCategory, IsFunction, IsInt, 83	for IsCapCategoryObject, IsCapCategory-
AddTensorUnit	Object, IsCapCategoryObject, IsCap-
for IsCapCategory, IsFunction, 98	CategoryObject, 8
for IsCapCategory, IsFunction, 18Int, 98	BraidingWithGivenTensorProducts
AddTraceMap	for IsCapCategoryObject, IsCapCategory-
for IsCapCategory, IsFunction, 101	Object, IsCapCategoryObject, IsCap-
	CategoryObject, 8
for IsCapCategory, IsFunction, IsInt, 101	2 7 3
AddUniversalPropertyOfCoDual	ClosedMonoidalCategoriesTest, 106
for IsCapCategory, IsFunction, 74	${\tt Closed Monoidal Categories Test With Given},$
for IsCapCategory, IsFunction, IsInt, 74	106
AddUniversalPropertyOfDual	${\tt Closed Monoidal Left Coevaluation Morphism}$
for IsCapCategory, IsFunction, 62	for IsCapCategoryObject, IsCapCategory-
for IsCapCategory, IsFunction, IsInt, 62	Object, 18
AddUniversalPropertyOfLeftCoDual	ClosedMonoidalLeftCoevaluation-
for IsCapCategory, IsFunction, 94	${\tt MorphismWithGivenRange}$
for IsCapCategory, IsFunction, IsInt, 94	for IsCapCategoryObject, IsCapCategory-
AddUniversalPropertyOfLeftDual	Object, IsCapCategoryObject, 18
for IsCapCategory, IsFunction, 84	ClosedMonoidalLeftEvaluationMorphism

for IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 116 Object, 18 CoclosedMonoidalCategoriesTest, 107 ClosedMonoidalLeftEvaluationMorphism-CoclosedMonoidalCategoriesTestWith-Given, 108 WithGivenSource for IsCapCategoryObject, IsCapCategory- ${\tt CoclosedMonoidalLeftCoevaluation-}$ Object, IsCapCategoryObject, 18 Morphism ClosedMonoidalRightCoevaluationfor IsCapCategoryObject, IsCapCategory-Object, 34 Morphism for IsCapCategoryObject, IsCapCategory- CoclosedMonoidalLeftCoevaluation-Object, 16 MorphismWithGivenSource ClosedMonoidalRightCoevaluationfor IsCapCategoryObject, IsCapCategory-MorphismWithGivenRange Object, IsCapCategoryObject, 34 for IsCapCategoryObject, IsCapCategory- CoclosedMonoidalLeftEvaluationMorphism Object, IsCapCategoryObject, 16 for IsCapCategoryObject, IsCapCategory- ${\tt ClosedMonoidalRightEvaluationMorphism}$ Object, 33 for IsCapCategoryObject, IsCapCategory-CoclosedMonoidalLeftEvaluation-Object, 16 MorphismWithGivenRange ClosedMonoidalRightEvaluationMorphismfor IsCapCategoryObject, IsCapCategory-WithGivenSource Object, IsCapCategoryObject, 34 CoclosedMonoidalRightCoevaluationfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 16 Morphism ${\tt CoclosedCoevaluationForCoDual}$ for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, 44 Object, 32 CoclosedCoevaluationForCoDualWith-CoclosedMonoidalRightCoevaluation-MorphismWithGivenSource GivenTensorProduct for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 44 Object, IsCapCategoryObject, 32 ${\tt CoclosedCoevaluationMorphism}$ CoclosedMonoidalRightEvaluationfor IsCapCategoryObject, IsCapCategory-Morphism for IsCapCategoryObject, IsCapCategory-Object, 116 CoclosedCoevaluationMorphismWithGiven-Object, 32 CoclosedMonoidalRightEvaluation-MorphismWithGivenRange for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 116 ${\tt CoclosedEvaluationForCoDual}$ Object, IsCapCategoryObject, 32 for IsCapCategoryObject, 36 CoDualityTensorProductCompatibility-CoclosedEvaluationForCoDualWithGiven-Morphism for IsCapCategoryObject, IsCapCategory-TensorProduct for IsCapCategoryObject, IsCapCategory-Object, 37 Object, IsCapCategoryObject, 36 CoDualityTensorProductCompatibility- ${\tt CoclosedEvaluationMorphism}$ MorphismWithGivenObjects for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, IsCap-Object, 116 CoclosedEvaluationMorphismWithGiven-CategoryObject, 37 Range CoDualOnMorphisms for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryMorphism, 36

${\tt CoDualOnMorphismsWithGivenCoDuals}$	InternalCoHom
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryCell, IsCapCategoryCell,
Morphism, IsCapCategoryObject, 36	45
CoDualOnObjects	InternalCoHomOnMorphisms
for IsCapCategoryObject, 36	for IsCapCategoryMorphism, IsCapCatego-
CoevaluationForDual	ryMorphism, 31
for IsCapCategoryObject, 41	InternalCoHomOnMorphismsWithGiven-
CoevaluationForDualWithGivenTensor-	InternalCoHoms
Product	for IsCapCategoryObject, IsCapCategory-
for IsCapCategoryObject, IsCapCategory-	Morphism, IsCapCategoryMorphism,
Object, IsCapCategoryObject, 41	IsCapCategoryObject, 32
CoevaluationMorphism	InternalCoHomOnObjects
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory-
Object, 117	Object, 31
CoevaluationMorphismWithGivenRange	InternalCoHomTensorProduct-
for IsCapCategoryObject, IsCapCategory-	${\tt CompatibilityMorphism}$
Object, IsCapCategoryObject, 117	for IsList, 37
CoLambdaElimination	InternalCoHomTensorProduct-
for IsCapCategoryObject, IsCapCategory-	${\tt CompatibilityMorphismInverse}$
Object, IsCapCategoryMorphism, 39	for IsList, 43
CoLambdaIntroduction	InternalCoHomTensorProduct-
for IsCapCategoryMorphism, 39	CompatibilityMorphismInverse-
CoRankMorphism	WithGivenObjects
for IsCapCategoryObject, 44	for IsCapCategoryObject, IsList, IsCapCate-
CoTraceMap	goryObject, 43
for IsCapCategoryMorphism, 44	InternalCoHomTensorProduct-
	CompatibilityMorphismWith-
DualOnMorphisms	GivenObjects
for IsCapCategoryMorphism, 21	for IsCapCategoryObject, IsList, IsCapCate-
${\tt DualOnMorphismsWithGivenDuals}$	goryObject, 37
for IsCapCategoryObject, IsCapCategory-	InternalCoHomToTensorProduct-
Morphism, IsCapCategoryObject, 21	AdjunctionMap
DualOnObjects	for IsObject, 117
for IsCapCategoryObject, 21	InternalCoHomToTensorProduct-
	${\tt AdjunctionMapWithGivenTensor-}$
EvaluationForDual	Product
for IsCapCategoryObject, 21	for IsObject, 117
EvaluationForDualWithGivenTensor-	InternalCoHomToTensorProductLeft-
Product	AdjunctionMap
for IsCapCategoryObject, IsCapCategory-	for IsObject, 118
Object, IsCapCategoryObject, 21	InternalCoHomToTensorProductLeft-
EvaluationMorphism	AdjunctMorphism
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory-
Object, 117	Object, IsCapCategoryMorphism, 35
EvaluationMorphismWithGivenSource	InternalCoHomToTensorProductLeft-
for IsCapCategoryObject, IsCapCategory-	AdjunctMorphismWithGivenTensor-
Object, IsCapCategoryObject, 117	

	Product	AdjunctionMap
for	IsCapCategoryObject, IsCapCategory-	for IsObject, 119
	Object, IsCapCategoryMorphism, Is-	InternalHomToTensorProductLeftAdjunct-
	CapCategoryObject, 35	Morphism
Interna	alCoHomToTensorProductRight-	for IsCapCategoryObject, IsCapCategory-
	AdjunctMorphism	Object, IsCapCategoryMorphism, 19
for	IsCapCategoryObject, IsCapCategory-	InternalHomToTensorProductLeftAdjunct-
	Object, IsCapCategoryMorphism, 33	${\tt MorphismWithGivenTensorProduct}$
Interna	alCoHomToTensorProductRight-	for IsCapCategoryObject, IsCapCategory-
	AdjunctMorphismWithGivenTensor-	Object, IsCapCategoryMorphism, Is-
	Product	CapCategoryObject, 19
for	IsCapCategoryObject, IsCapCategory-	${\tt Internal HomToTensor Product Right-}$
	Object, IsCapCategoryMorphism, Is-	${\tt AdjunctionIsomorphism}$
	CapCategoryObject, 33	for IsCapCategoryObject, IsCapCategory-
Interna	alHom	Object, IsCapCategoryObject, 17
for	IsCapCategoryCell, IsCapCategoryCell,	${\tt Internal HomToTensor Product Right-}$
	45	${\tt Adjunction Isomorphism With-}$
	alHomOnMorphisms	GivenObjects
for	IsCapCategoryMorphism, IsCapCatego-	for IsCapCategoryObject, IsCapCategory-
	ryMorphism, 15	Object, IsCapCategoryObject, IsCap-
Interna	alHomOnMorphismsWithGiven-	CategoryObject, IsCapCategoryObject,
_	InternalHoms	18
for	IsCapCategoryObject, IsCapCategory-	InternalHomToTensorProductRight-
	Morphism, IsCapCategoryMorphism,	${\tt AdjunctMorphism}$
	IsCapCategoryObject, 15	for IsCapCategoryObject, IsCapCategory-
	alHomOnObjects	Object, IsCapCategoryMorphism, 17
tor	IsCapCategoryObject, IsCapCategory-	InternalHomToTensorProductRight-
- .	Object, 15	AdjunctMorphismWithGivenTensor-
Interna	alHomToTensorProduct-	Product
c ,	AdjunctionMap	for IsCapCategoryObject, IsCapCategory-
	IsObject, 118	Object, IsCapCategoryMorphism, Is-
Interna	alHomToTensorProductAdjunction-	CapCategoryObject, 17
c ,	MapWithGivenTensorProduct	IsBraidedMonoidalCategory
	IsObject, 118	for IsCapCategory, 120
Interna	alHomToTensorProductLeft-	IsClosedMonoidalCategory
for	AdjunctionIsomorphism	for IsCapCategory, 120
101	IsCapCategoryObject, IsCapCategory-Object IsCapCategoryObject 20	IsCoclosedMonoidalCategory
T., +	Object, IsCapCategoryObject, 20 alHomToTensorProductLeft-	for IsCapCategory, 120
Interna		IsLeftClosedMonoidalCategory
	AdjunctionIsomorphismWith-	for IsCapCategory, 120
for	GivenObjects	IsLeftCoclosedMonoidalCategory
101	IsCapCategoryObject, IsCapCategory-	for IsCapCategory, 120
	Object, IsCapCategoryObject, IsCap-	IsMonoidalCategory
	CategoryObject, IsCapCategoryObject, 20	for IsCapCategory, 121
Tn+ ~~~	alHomToTensorProductLeft-	IsomorphismFromCoDualObjectToInternal- CoHomFromTensorUnit
THELUS	arnomiolensorrroducthert-	Conomriomiensorunit

for IsCapCategoryObject, 38	Object, 31
IsomorphismFromDualObjectToInternal-	IsomorphismFromLeftInternalHomInto-
HomIntoTensorUnit	${\tt TensorUnitToLeftDualObject}$
for IsCapCategoryObject, 23	for IsCapCategoryObject, 14
IsomorphismFromInternalCoHomFrom-	IsomorphismFromLeftInternalHomToObject
TensorUnitToCoDualObject	for IsCapCategoryObject, 15
for IsCapCategoryObject, 38	IsomorphismFromLeftInternalHomTo-
IsomorphismFromInternalCoHomToObject	ObjectWithGivenLeftInternalHom
for IsCapCategoryObject, 39	for IsCapCategoryObject, IsCapCategory
IsomorphismFromInternalCoHomToObject-	Object, 15
WithGivenInternalCoHom	IsomorphismFromObjectToInternalCoHom
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, 39
Object, 39	IsomorphismFromObjectToInternalCoHom-
IsomorphismFromInternalCoHomToTensor-	WithGivenInternalCoHom
ProductWithCoDualObject	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, IsCapCategory-	Object, 39
Object, 42	IsomorphismFromObjectToInternalHom
IsomorphismFromInternalHomIntoTensor-	for IsCapCategoryObject, 24
UnitToDualObject	IsomorphismFromObjectToInternalHom-
for IsCapCategoryObject, 23	WithGivenInternalHom
IsomorphismFromInternalHomToObject	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, 24	Object, 24
IsomorphismFromInternalHomToObject-	IsomorphismFromObjectToLeftInternal-
WithGivenInternalHom	CoHom
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, 30
Object, 24	IsomorphismFromObjectToLeftInternalCo-
IsomorphismFromInternalHomToTensor-	${\tt HomWithGivenLeftInternalCoHom}$
ProductWithDualObject	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, IsCapCategory-	Object, 31
Object, 40	IsomorphismFromObjectToLeftInternalHom
IsomorphismFromLeftCoDualObjectToLeft-	for IsCapCategoryObject, 14
InternalCoHomFromTensorUnit	IsomorphismFromObjectToLeftInternal-
for IsCapCategoryObject, 30	HomWithGivenLeftInternalHom
IsomorphismFromLeftDualObjectToLeft-	for IsCapCategoryObject, IsCapCategory
InternalHomIntoTensorUnit	Object, 14
for IsCapCategoryObject, 13	IsomorphismFromTensorProductWithCo-
IsomorphismFromLeftInternalCoHomFrom-	DualObjectToInternalCoHom
TensorUnitToLeftCoDualObject	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, 30	Object, 43
IsomorphismFromLeftInternalCoHomTo-	IsomorphismFromTensorProductWithDual-
Object	ObjectToInternalHom
for IsCapCategoryObject, 31	for IsCapCategoryObject, IsCapCategory
IsomorphismFromLeftInternalCoHomTo-	Object, 40
ObjectWithGivenLeftInternal-	IsRigidSymmetricClosedMonoidalCategory
CoHom	for IsCapCategory, 121
for IsCapCategoryObject, IsCapCategory-	

Category	for IsCapCategoryObject, IsCapCategory-
for IsCapCategory, 121	Object, IsCapCategoryObject, 11
IsStrictMonoidalCategory	${\tt LeftClosedMonoidalPostComposeMorphism-}$
for IsCapCategory, 121	WithGivenObjects
IsSymmetricClosedMonoidalCategory	for IsCapCategoryObject, IsCapCategory-
for IsCapCategory, 121	Object, IsCapCategoryObject, IsCap-
IsSymmetricCoclosedMonoidalCategory	CategoryObject, IsCapCategoryObject,
for IsCapCategory, 121	11
IsSymmetricMonoidalCategory	LeftClosedMonoidalPreComposeMorphism
for IsCapCategory, 121	for IsCapCategoryObject, IsCapCategory-
	Object, IsCapCategoryObject, 10
LambdaElimination	LeftClosedMonoidalPreComposeMorphism-
for IsCapCategoryObject, IsCapCategory-	WithGivenObjects
Object, IsCapCategoryMorphism, 24	for IsCapCategoryObject, IsCapCategory-
LambdaIntroduction	Object, IsCapCategoryObject, IsCap-
for IsCapCategoryMorphism, 23	CategoryObject, IsCapCategoryObject,
LeftClosedMonoidalCategoriesTest, 108	11
LeftClosedMonoidalCategoriesTestWith-	${\tt LeftCoclosedMonoidalCategoriesTest,109}$
Given, 109	${\tt LeftCoclosedMonoidalCategoriesTest-}$
LeftClosedMonoidalCoevaluationMorphism	WithGiven, 110
for IsCapCategoryObject, IsCapCategory-	${\tt LeftCoclosedMonoidalCoevaluation-}$
Object, 9	Morphism
LeftClosedMonoidalCoevaluation-	for IsCapCategoryObject, IsCapCategory-
MorphismWithGivenRange	Object, 25
for IsCapCategoryObject, IsCapCategory-	LeftCoclosedMonoidalCoevaluation-
Object, IsCapCategoryObject, 9 LeftClosedMonoidalEvaluationForLeft-	${\tt MorphismWithGivenSource}$
Dual	for IsCapCategoryObject, IsCapCategory-
for IsCapCategoryObject, 12	Object, IsCapCategoryObject, 26
LeftClosedMonoidalEvaluationForLeft-	LeftCoclosedMonoidalEvaluationForLeft-
DualWithGivenTensorProduct	CoDual
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, 28
Object, IsCapCategoryObject, 12	LeftCoclosedMonoidalEvaluationForLeft-
LeftClosedMonoidalEvaluationMorphism	CoDualWithGivenTensorProduct
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory-
Object, 9	Object, IsCapCategoryObject, 28
LeftClosedMonoidalEvaluationMorphism-	LeftCoclosedMonoidalEvaluationMorphism
WithGivenSource	for IsCapCategoryObject, IsCapCategory-
for IsCapCategoryObject, IsCapCategory-	Object, 25
Object, IsCapCategoryObject, 9	LeftCoclosedMonoidalEvaluation-
LeftClosedMonoidalLambdaElimination	MorphismWithGivenRange
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory-
Object, IsCapCategoryMorphism, 14	Object, IsCapCategoryObject, 25
LeftClosedMonoidalLambdaIntroduction	LeftCoclosedMonoidalLambdaElimination
for IsCapCategoryMorphism, 14	for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 30
LeftClosedMonoidalPostComposeMorphism	LeftCoclosedMonoidalLambdaIntroduction
1 1	rer cooctobedionotaatrampaatuttoancitou

for IsCapCategoryMorphism, 30	for IsCapCategoryObject, IsList, 6
LeftCoclosedMonoidalPostCoCompose-	LeftDistributivityFactoringWithGiven-
Morphism	Objects
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory-
Object, IsCapCategoryObject, 27	Object, IsList, IsCapCategoryObject, 6
LeftCoclosedMonoidalPostCoCompose-	LeftDualOnMorphisms
MorphismWithGivenObjects	for IsCapCategoryMorphism, 11
for IsCapCategoryObject, IsCapCategory-	LeftDualOnMorphismsWithGivenLeftDuals
Object, IsCapCategoryObject, IsCap-	for IsCapCategoryObject, IsCapCategory-
CategoryObject, IsCapCategoryObject,	Morphism, IsCapCategoryObject, 11
27	
	LeftDualOnObjects
LeftCoclosedMonoidalPreCoCompose-	for IsCapCategoryObject, 11
Morphism	LeftInternalCoHom
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryCell, IsCapCategoryCell,
Object, IsCapCategoryObject, 27	45
LeftCoclosedMonoidalPreCoCompose-	LeftInternalCoHomOnMorphisms
${\tt MorphismWithGivenObjects}$	for IsCapCategoryMorphism, IsCapCatego-
for IsCapCategoryObject, IsCapCategory-	ryMorphism, 25
Object, IsCapCategoryObject, IsCap-	${\tt LeftInternalCoHomOnMorphismsWithGiven-}$
CategoryObject, IsCapCategoryObject,	${ t LeftInternalCoHoms}$
27	for IsCapCategoryObject, IsCapCategory-
LeftCoDualityTensorProduct-	Morphism, IsCapCategoryMorphism,
${\tt CompatibilityMorphism}$	IsCapCategoryObject, 25
for IsCapCategoryObject, IsCapCategory-	LeftInternalCoHomOnObjects
	for IsCapCategoryObject, IsCapCategory-
Object, 29	101 100 00 00 00 00 00 00 00 00 00 00 00
	Object, 25
LeftCoDualityTensorProduct-	
LeftCoDualityTensorProduct- CompatibilityMorphismWith-	Object, 25 LeftInternalCoHomTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory-	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap-	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCate-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo-	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCate- goryObject, 29
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCate- goryObject, 29 LeftInternalCoHomToTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory-	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCate- goryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCate- goryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26 LeftInternalCoHomToTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26 LeftInternalCoHomToTensorProduct- AdjunctMorphismWithGivenTensor-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding for IsCapCategoryObject, IsList, 6	Object, 25 LeftInternalCoHomTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding for IsCapCategoryObject, IsList, 6 LeftDistributivityExpandingWithGiven-	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26 LeftInternalCoHomToTensorProduct- AdjunctMorphismWithGivenTensor- Product for IsCapCategoryObject, IsCapCategory-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding for IsCapCategoryObject, IsList, 6 LeftDistributivityExpandingWithGiven- Objects	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26 LeftInternalCoHomToTensorProduct- AdjunctMorphismWithGivenTensor- Product for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, Is-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding for IsCapCategoryObject, IsList, 6 LeftDistributivityExpandingWithGiven- Objects for IsCapCategoryObject, IsCapCategory-	Object, 25 LeftInternalCoHomTensorProduct-
LeftCoDualityTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryObject, IsCap- CategoryObject, 29 LeftCoDualOnMorphisms for IsCapCategoryMorphism, 28 LeftCoDualOnMorphismsWithGivenLeftCo- Duals for IsCapCategoryObject, IsCapCategory- Morphism, IsCapCategoryObject, 28 LeftCoDualOnObjects for IsCapCategoryObject, 27 LeftDistributivityExpanding for IsCapCategoryObject, IsList, 6 LeftDistributivityExpandingWithGiven- Objects	Object, 25 LeftInternalCoHomTensorProduct- CompatibilityMorphism for IsList, 28 LeftInternalCoHomTensorProduct- CompatibilityMorphismWith- GivenObjects for IsCapCategoryObject, IsList, IsCapCategoryObject, 29 LeftInternalCoHomToTensorProduct- AdjunctMorphism for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, 26 LeftInternalCoHomToTensorProduct- AdjunctMorphismWithGivenTensor- Product for IsCapCategoryObject, IsCapCategory- Object, IsCapCategoryMorphism, Is-

45 for IsCapCategoryObject, IsCapCategory-LeftInternalHomOnMorphisms Object, IsCapCategoryObject, 20 for IsCapCategoryMorphism, IsCapCatego- MonoidalPostComposeMorphismWithGivenryMorphism, 8 Objects LeftInternalHomOnMorphismsWithGivenfor IsCapCategoryObject, IsCapCategory-LeftInternalHoms Object, IsCapCategoryObject, IsCap-CategoryObject, IsCapCategoryObject, for IsCapCategoryObject, IsCapCategory-IsCapCategoryMorphism, Morphism, 21 IsCapCategoryObject, 9 ${\tt MonoidalPreCoComposeMorphism}$ LeftInternalHomOnObjects for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, 35 MonoidalPreCoComposeMorphismWithGiven-Object, 8 LeftInternalHomToTensorProductAdjunct-Objects Morphism for IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryObject, IsCapfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryMorphism, 10 CategoryObject, IsCapCategoryObject, LeftInternalHomToTensorProductAdjunct- ${\tt MorphismWithGivenTensorProduct}$ ${\tt MonoidalPreComposeMorphism}$ for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, IsCapCategoryMorphism, Is-Object, IsCapCategoryObject, 20 CapCategoryObject, 10 MonoidalPreComposeMorphismWithGiven-LeftUnitor Objects for IsCapCategoryObject, 4 for IsCapCategoryObject, IsCapCategory-LeftUnitorInverse Object, IsCapCategoryObject, IsCap-CategoryObject, IsCapCategoryObject, for IsCapCategoryObject, 5 LeftUnitorInverseWithGivenTensor-20 Product ${\tt MorphismFromBidual}$ for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, 42 MorphismFromBidualWithGivenBidual Object, 5 ${\tt LeftUnitorWithGivenTensorProduct}$ for IsCapCategoryObject, IsCapCategoryfor IsCapCategoryObject, IsCapCategory-Object, 42 Object, 5 MorphismFromCoBidual for IsCapCategoryObject, 37 ${\tt MonoidalCategoriesTensorProductOn-}$ ${\tt MorphismFromCoBidualWithGivenCoBidual}$ ObjectsAndTensorUnitTest, 110 for IsCapCategoryObject, IsCapCategory-MonoidalCategoriesTest, 111 Object, 37 ${\tt MonoidalPostCoComposeMorphism}$ MorphismFromInternalCoHomToTensorfor IsCapCategoryObject, IsCapCategory-Product Object, IsCapCategoryObject, 35 for IsCapCategoryObject, IsCapCategory- ${\tt MonoidalPostCoComposeMorphismWith-}$ Object, 38 GivenObjects MorphismFromInternalCoHomToTensorfor IsCapCategoryObject, IsCapCategory-ProductWithGivenObjects Object, IsCapCategoryObject, IsCapfor IsCapCategoryObject, IsCapCategory-CategoryObject, IsCapCategoryObject, Object, IsCapCategoryObject, IsCap-CategoryObject, 38

 ${\tt MorphismFromInternalHomToTensorProduct}$

MonoidalPostComposeMorphism

for	IsCapCategoryObject, IsCapCategory-	Object, IsCapCategoryObject, IsCap-
	Object, 40	CategoryObject, 13
Morphis	smFromInternalHomToTensor-	MorphismToBidual
	ProductWithGivenObjects	for IsCapCategoryObject, 21
for	IsCapCategoryObject, IsCapCategory-	${\tt MorphismToBidualWithGivenBidual}$
	Object, IsCapCategoryObject, IsCap-	for IsCapCategoryObject, IsCapCategory-
	CategoryObject, 41	Object, 22
Morphis	smFromLeftCoBidual	MorphismToCoBidual
for	IsCapCategoryObject, 28	for IsCapCategoryObject, 44
Morphis	smFromLeftCoBidualWithGivenLeft-	${\tt MorphismToCoBidualWithGivenCoBidual}$
	CoBidual	for IsCapCategoryObject, IsCapCategory-
for	IsCapCategoryObject, IsCapCategory-	Object, 44
	Object, 28	MorphismToLeftBidual
Morphis	smFromLeftInternalCoHomToTensor-	for IsCapCategoryObject, 12
	Product	MorphismToLeftBidualWithGivenLeft-
for	IsCapCategoryObject, IsCapCategory-	Bidual
	Object, 29	for IsCapCategoryObject, IsCapCategory-
Morphis	smFromLeftInternalCoHomToTensor-	Object, 12
_	ProductWithGivenObjects	•
for	IsCapCategoryObject, IsCapCategory-	RankMorphism
	Object, IsCapCategoryObject, IsCap-	for IsCapCategoryObject, 42
	CategoryObject, 29	$ ext{RightDistributivityExpanding}$
Morphis	smFromTensorProductToInternal-	for IsList, IsCapCategoryObject, 7
-	CoHom	RightDistributivityExpandingWithGiven-
for	IsCapCategoryObject, IsCapCategory-	Objects
	Object, 43	for IsCapCategoryObject, IsList, IsCapCate-
Morphis	smFromTensorProductToInternalCo-	goryObject, IsCapCategoryObject, 7
-	HomWithGivenObjects	$ ext{RightDistributivityFactoring}$
for	IsCapCategoryObject, IsCapCategory-	for IsList, IsCapCategoryObject, 7
	Object, IsCapCategoryObject, IsCap-	RightDistributivityFactoringWithGiven-
	CategoryObject, 43	Objects
Morphis	smFromTensorProductToInternalHom	for IsCapCategoryObject, IsList, IsCapCate-
_	IsCapCategoryObject, IsCapCategory-	goryObject, IsCapCategoryObject, 7
	Object, 23	RightUnitor
Morphis	smFromTensorProductToInternal-	for IsCapCategoryObject, 5
1	HomWithGivenObjects	RightUnitorInverse
for	IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, 5
	Object, IsCapCategoryObject, IsCap-	RightUnitorInverseWithGivenTensor-
	CategoryObject, 23	Product
Morphis	smFromTensorProductToLeft-	for IsCapCategoryObject, IsCapCategory-
P	InternalHom	Object, 5
for	IsCapCategoryObject, IsCapCategory-	RightUnitorWithGivenTensorProduct
101	Object, 13	for IsCapCategoryObject, IsCapCategory-
Morphis	smFromTensorProductToLeft-	Object, 5
P	InternalHomWithGivenObjects	RigidSymmetricClosedMonoidal-
for	IsCapCategoryObject, IsCapCategory-	CategoriesTest, 111
101	is cap category coject, is cap category	

RigidSymmetricCoclosedMonoidal-	TensorProductUnMorphismsWithGiven-
CategoriesTest, 112	TensorProducts
T	for IsCapCategoryObject, IsCapCategory-
TensorProductDualityCompatibility-	Morphism, IsCapCategoryMorphism,
Morphism	IsCapCategoryObject, 4
for IsCapCategoryObject, IsCapCategory-	TensorProductOnObjects
Object, 22	for IsCapCategoryObject, IsCapCategory-
TensorProductDualityCompatibility-	Object, 6
MorphismWithGivenObjects	TensorProductToInternalCoHom-
for IsCapCategoryObject, IsCapCategory-	${\tt AdjunctionMap}$
Object, IsCapCategoryObject, IsCap-	for IsObject, 118
CategoryObject, 22	TensorProductToInternalCoHom-
TensorProductInternalHomCompatibility-	${\tt AdjunctionMapWithGivenInternal-}$
Morphism	CoHom
for IsList, 22	for IsObject, 118
TensorProductInternalHomCompatibility-	TensorProductToInternalCoHomLeft-
${\tt MorphismInverse}$	${\tt AdjunctionMap}$
for IsList, 41	for IsObject, 119
TensorProductInternalHomCompatibility-	TensorProductToInternalCoHomLeft-
${\tt MorphismInverseWithGivenObjects}$	${\tt AdjunctionMapWithGivenInternal-}$
for IsCapCategoryObject, IsList, IsCapCate-	CoHom
goryObject, 41	for IsObject, 119
${\tt TensorProductInternalHomCompatibility-}$	TensorProductToInternalCoHomLeft-
$ exttt{MorphismWithGivenObjects}$	${\tt AdjunctMorphism}$
for IsCapCategoryObject, IsList, IsCapCate-	for IsCapCategoryObject, IsCapCategory-
goryObject, 22	Object, IsCapCategoryMorphism, 34
${\tt TensorProductLeftDualityCompatibility-}$	TensorProductToInternalCoHomLeft-
Morphism	AdjunctMorphismWithGiven-
for IsCapCategoryObject, IsCapCategory-	InternalCoHom
Object, 13	for IsCapCategoryObject, IsCapCategory-
TensorProductLeftDualityCompatibility-	Object, IsCapCategoryMorphism, Is-
MorphismWithGivenObjects	CapCategoryObject, 34
for IsCapCategoryObject, IsCapCategory-	TensorProductToInternalCoHomRight-
Object, IsCapCategoryObject, IsCap-	AdjunctMorphism
CategoryObject, 13	for IsCapCategoryObject, IsCapCategory-
TensorProductLeftInternalHom-	Object, IsCapCategoryMorphism, 33
${\tt CompatibilityMorphism}$	TensorProductToInternalCoHomRight-
for IsList, 12	AdjunctMorphismWithGiven-
TensorProductLeftInternalHom-	InternalCoHom
CompatibilityMorphismWith-	for IsCapCategoryObject, IsCapCategory-
GivenObjects	Object, IsCapCategoryMorphism, Is-
for IsCapCategoryObject, IsList, IsCapCate-	CapCategoryObject, 33
goryObject, 12	TensorProductToInternalHom-
TensorProductOnMorphisms	AdjunctionMap
for IsCapCategoryMorphism, IsCapCatego-	for IsObject, 118
ryMorphism, 3	TensorProductToInternalHomAdiunction-
▼ •	- 1 CTD O 1 1 T O 4 4 C C T O 1 D 1 T O C 1 T O C 1 T O C T T O C T T O C T T O C T T O C T T O C T O

${ t MapWithGivenInternalHom}$	InternalHom
for IsObject, 118	for IsCapCategoryObject, IsCapCategory
TensorProductToInternalHomLeft-	Object, IsCapCategoryMorphism, Is
${\tt AdjunctionIsomorphism}$	CapCategoryObject, 16
for IsCapCategoryObject, IsCapCategory-	TensorProductToLeftInternalCoHom-
Object, IsCapCategoryObject, 19	AdjunctMorphism
TensorProductToInternalHomLeft-	for IsCapCategoryObject, IsCapCategory
AdjunctionIsomorphismWith-	Object, IsCapCategoryMorphism, 26
GivenObjects	TensorProductToLeftInternalCoHom-
for IsCapCategoryObject, IsCapCategory-	AdjunctMorphismWithGivenLeft-
Object, IsCapCategoryObject, IsCap-	InternalCoHom
CategoryObject, IsCapCategoryObject,	for IsCapCategoryObject, IsCapCategory
19	Object, IsCapCategoryMorphism, Is
TensorProductToInternalHomLeft-	CapCategoryObject, 26
${\tt AdjunctionMap}$	TensorProductToLeftInternalHomAdjunct-
for IsObject, 119	Morphism
TensorProductToInternalHomLeft-	for IsCapCategoryObject, IsCapCategory
AdjunctionMapWithGiven-	Object, IsCapCategoryMorphism, 10
InternalHom	TensorProductToLeftInternalHom-
for IsObject, 119	${\tt AdjunctMorphismWithGivenLeft-}$
${\tt TensorProductToInternalHomLeftAdjunct-}$	InternalHom
Morphism	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, IsCapCategory-	Object, IsCapCategoryMorphism, Is
Object, IsCapCategoryMorphism, 18	CapCategoryObject, 10
${\tt TensorProductToInternalHomLeftAdjunct-}$	TensorUnit
${\tt MorphismWithGivenInternalHom}$	for IsCapCategory, 6
for IsCapCategoryObject, IsCapCategory-	TraceMap
Object, IsCapCategoryMorphism, Is-	for IsCapCategoryMorphism, 41
CapCategoryObject, 19	
TensorProductToInternalHomRight-	UniversalPropertyOfCoDual
${\tt AdjunctionIsomorphism}$	for IsCapCategoryObject, IsCapCategory
for IsCapCategoryObject, IsCapCategory-	Object, IsCapCategoryMorphism, 38
Object, IsCapCategoryObject, 17	UniversalPropertyOfDual
TensorProductToInternalHomRight-	for IsCapCategoryObject, IsCapCategory
AdjunctionIsomorphismWith-	Object, IsCapCategoryMorphism, 23
GivenObjects	UniversalPropertyOfLeftCoDual
for IsCapCategoryObject, IsCapCategory-	for IsCapCategoryObject, IsCapCategory
Object, IsCapCategoryObject, IsCap-	Object, IsCapCategoryMorphism, 30
CategoryObject, IsCapCategoryObject,	UniversalPropertyOfLeftDual
17	for IsCapCategoryObject, IsCapCategory
${\tt TensorProductToInternalHomRight-}$	Object, IsCapCategoryMorphism, 14
${\tt AdjunctMorphism}$	WriteFileForClosedMonoidalStructure,
for IsCapCategoryObject, IsCapCategory-	113
Object, IsCapCategoryMorphism, 16	WriteFileForCoclosedMonoidalStructure,
${\tt TensorProductToInternalHomRight-}$	114
AdjunctMorphismWithGiven-	117

WriteFileForLeftClosedMonoidal-Structure, 113 WriteFileForLeftCoclosedMonoidal-Structure, 114 WriteFileForMonoidalStructure, 113