Category Theory with Strings

Shunsuke Sogame

March 19, 2016

1 Introduction

This is a complementary document for introductory books of category theory ([1], [2], [3], [8]) using *string diagrams*. Don't trust my poor mathematics. Any feedback is welcome at github.com/okomok/strcat.

2 Preliminaries

2.1 Universality

Definition 2.1 For a boolean-valued function P, define

$$!aP(a) \coloneqq P(a) \wedge \forall a'(P(a') \implies a = a')$$

Definition 2.2 (Uniqueness Quantification) Define

$$\exists! aP(a) \coloneqq \exists a!aP(a)$$

meaning that "there exists a unique a such that P".

Remark 2.3 On the other hand,

$$\exists a\big((!aP(a)) \wedge Q(a)\big)$$

means "there exists a unique a such that P, moreover the a is Q".

Definition 2.4 (Universality) Given a binary boolean-valued function P, we boldly call a statement of the form

$$(\forall x \in X)(\exists! y \in Y)(P(x, y))$$

the *universality* of P.

Proposition 2.5 (Functional Universality)

$$\begin{aligned} &(\forall x \in X)(\exists! y \in Y)(P(x, y)) \\ \iff &(\exists f : X \to Y)(\forall x \in X)(\forall y \in Y)(P(x, y) \iff y = f(x)) \end{aligned}$$

PROOF. ($\implies$) by the Axiom of choice. ($\impliedby$) immediate. ⌑

Definition 2.6 (Bijectivity) Given a a function $g : Y \to X$, a statement

$$(\exists f : X \to Y)(\forall x \in X)(\forall y \in Y)(x = g(y) \iff y = f(x))$$

is known as the *bijectivity* of f and g.

This is a special case of universality where $P(x, y)$ is $x = g(y)$.

2.2 Lambda Expressions

Definition 2.7 (Lambda Expression) Following famous symbols like Σ, define

$$\Lambda_x y \coloneqq x \mapsto y$$

for anonymous functions.

Definition 2.8 Given a function H whose domain is a set of functions, define

$$H_x y \coloneqq H(\Lambda_x y)$$

Definition 2.9 (Placeholder Expression) For simple lambda expressions, you may use *placeholders*:

$$\texttt{?}{+}1 \coloneqq \Lambda_n n{+}1$$

Placeholder symbols can vary: `?`, `-`, `1`, etc.

2.3 Families

Syntax of the function application is world-standard and fixed:

$$f(x) \text{ or } fx$$

but sometimes you might want cuter syntax like that

$$\langle\!\langle x \rangle\!\rangle$$

Definition 2.10 (Family Declaration) A *family declaration* is an easy way to define arbitary application syntax:

$$(\langle\!\langle x \rangle\!\rangle \in Y)_{x \in X}$$

Definition 2.11 (Family Definition) If you place a function implementation into a family declaration:

$$(\langle\!\langle x \rangle\!\rangle \coloneqq x^2 \in Y)_{x \in X}$$

it might be called a *family definition*.

Family declarations can do more.

Definition 2.12 (Dependent Function) Let F a set-valued function.

$$(f(x) \in F(x))_{x \in X}$$

defines a function

$$f : X \to \bigcup_{x \in X} F(x)$$

such that

$$(\forall x \in X)(f(x) \in F(x))$$

Such f is called a *dependent function*, for the $F(x)$ depends on x.

In case F is a constant function, f is a normal function $X \to Y$.

2.4 Coherence

You will write

$$3 + 1 + 2$$

rather than

$$(3 + (0 + 1)) + 2$$

because you know the arithmetic laws

$$\begin{aligned} x + (y + z) &= (x + y) + z \\ 0 + x &= x = x + 0 \end{aligned}$$

disambiguate unparenthesized expressions. Informally laws to introduce simpler syntax are called *coherence conditions* or briefly *coherence*.

3 Categories

3.1 The Definition

Definition 3.1 (Category) A *category* $\mathcal{C}$ consists of

1. *objects*: a class $\mathrm{Ob}(\mathcal{C})$
2. *morphisms* or *hom-sets*: a family of sets $(\mathcal{C}(A,B))_{A,B\in\mathrm{Ob}(\mathcal{C})}$
3. *compositions*: a family of functions
$$(\circ : \mathcal{C}(B,C)\times\mathcal{C}(A,B)\to\mathcal{C}(A,C))_{A,B,C\in\mathrm{Ob}(\mathcal{C})}$$
4. *idenitities* or *units*: a family of morphisms
$$(\mathrm{id}_A\in\mathcal{C}(A,A))_{A\in\mathrm{Ob}(\mathcal{C})}$$

satisfying the following coherence conditions

1. *associativity*: for any $f\in\mathcal{C}(A,B)$, $g\in\mathcal{C}(B,C)$, and $h\in\mathcal{C}(C,D)$,
$$h\circ(g\circ f)=(h\circ g)\circ f$$
2. *unitality*: for any $f\in\mathcal{C}(A,B)$,
$$\mathrm{id}_B\circ f=f=f\circ\mathrm{id}_A$$

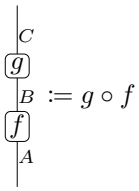
A morphism $f\in\mathcal{C}(A,B)$ is often denoted as $f:A\to B$.

3.2 String Diagrams

From now on, we will introduce *string diagrams* to complement(or hopefully replace) commutative diagrams, where an object A is depicted as an optionally-tagged string

A morphism $f\in\mathcal{C}(A,B)$ is depicted as a node

A composition joins two strings:



$\coloneqq g \circ f$

An idenity is indistinguishable from an object:

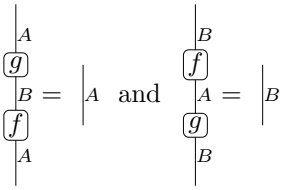
$\coloneqq \mathrm{id}_A$

Check these diagrams create no ambiguity thanks to the coherence.

Definition 3.2 (Isomorphism) An *isomorphism* is a pair of morphisms

$$f : A \to B$$
$$g : B \to A$$

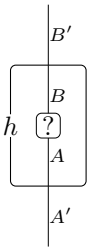
satisfying the *invertibility*



Definition 3.3 (Functorial Box) Given categories $\mathcal{C}$ and $\mathcal{C}'$, a function

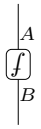
$$h : \mathcal{C}(A, B) \to \mathcal{C}'(A', B')$$

is depicted as



Definition 3.4 (Opposite Category) Given a category $\mathcal{C}$ and a morphism

you can build a category with strings upsidedown:



, which is denoted as $\mathcal{C}^{\mathrm{op}}$, the *opposite category* of $\mathcal{C}$.

Definition 3.5 (Discrete Category) A category $\mathcal{C}$ such that

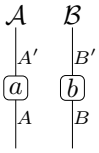
$$A = B \implies \mathcal{C}(A, B) = \{\mathrm{id}_A\}$$
$$A \neq B \implies \mathcal{C}(A, B) = \emptyset$$

is called a *discrete category*. Any set can be represented as a discrete category.

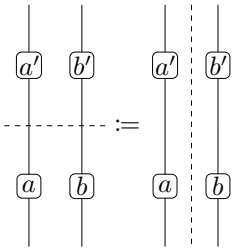
Definition 3.6 (Product Category) Given two categories $\mathcal{A}$ and $\mathcal{B}$, the *product category*

$$\mathcal{A} \times \mathcal{B}$$

is depicted as parallel strings

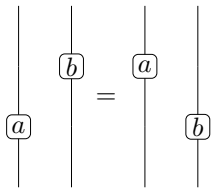


A composition, which joins parallel strings, is defined by



An identity is trivially

By these definitions,



4 Functors

4.1 The Definition

Definition 4.1 (Functor) A *functor* $F : \mathcal{C} \to \mathcal{D}$ consists of:

1. *domain*: a category $\mathcal{C}$
2. *codomain*: a category $\mathcal{D}$
3. a family of objects $(FA \in \mathrm{Ob}(\mathcal{D}))_{A \in \mathrm{Ob}(\mathcal{C})}$
4. families of morphisms

$$\big((F(f) \in \mathcal{D}(FA, FB))_{f \in \mathcal{C}(A,B)}\big)_{A,B \in \mathrm{Ob}(\mathcal{C})}$$

satisfying the *functoriality*:

1. *composition-compatibility*: for any $f \in \mathcal{C}(A, B)$ and $g \in \mathcal{C}(B, C)$,

$$F(g \circ f) = F(g) \circ F(f)$$

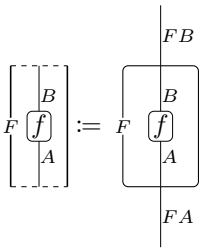
2. *unit-compatibility*: for any $A \in \mathrm{Ob}(\mathcal{C})$,

$$F(\mathrm{id}_A) = \mathrm{id}_{FA}$$

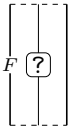
Definition 4.2 (Infrafunctor) An *infrafunctor* is a functor without the requirement of functoriality.

4.2 Functorial Tubes

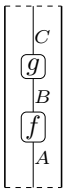
In string diagrams, a functor is represented as a tube



Placeholders make it simple:

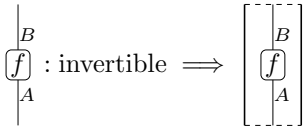


One can check the functoriality ensures any tube like



be unambiguous. "Join then tube" is the same as "Tube then join".

Proposition 4.3 Any functor preserves the invertibility meaning that



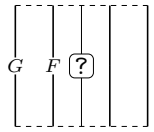
Definition 4.4 (Composite Functor) For any two functors

$$F : \mathcal{A} \to \mathcal{B}$$
$$G : \mathcal{B} \to \mathcal{C}$$

, the *composite functor* of F and G

$$G \circ F : \mathcal{A} \to \mathcal{C}$$

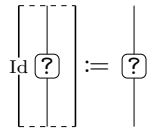
is depicted as



Definition 4.5 (Identity Functor) An *idenity functor*

$$\mathrm{Id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$$

is depicted as

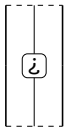


Definition 4.6 (Contravariant Functor) A functor whose domain is an opposite category

$$F : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$$

is called *contravariant*, while a normal functor is called *covariant*.

A contravariant functor is depicted as



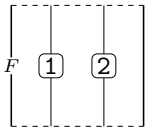
Definition 4.7 (Variant) Given a statement regarding functors, you can obtain a corresponding one regarding contravariant functors and vice versa. We call such a statement the *variant* of the original one.

Definition 4.8 (Binary Functor) A functor whose domain is a product category

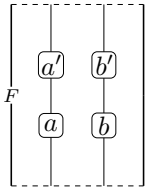
$$F : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$$

is called a *binary functor* or *bifunctor*.

With numbered placeholders, it is depicted as



Spelling out the definition of functoriality, one can check a diagram like

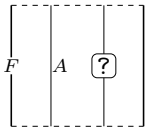


is unambiguous.

Definition 4.9 (Partial Application) Given a binary functor $F : \mathcal{A} \times \mathcal{B} \to \mathcal{C}$, a *partially applied* functor

$$\Lambda_B F(A, B) : \mathcal{B} \to \mathcal{C} \text{ or briefly}$$
$$F(A, \texttt{?}) : \mathcal{B} \to \mathcal{C}$$

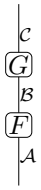
is defined by



The definition of $F(\texttt{?}, B)$ is an exercise.

Definition 4.10 (Small Category) A category $\mathcal{C}$ is called *small* when its $\mathrm{Ob}(\mathcal{C})$ is a set.

Definition 4.11 (Category of Small Categories) The *category of small categories* **Cat** is the category whose objects are all small categories and whose morphisms are functors:



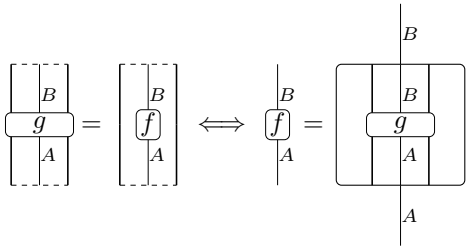
, where composite functors join the strings.

Definition 4.12 (Full and Faithful Functor) A functor $F : \mathcal{C} \to \mathcal{D}$ is called *full and faithful* if for each object A and B in $\mathcal{C}$, the family

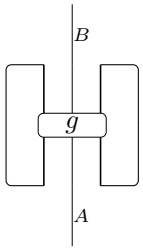
$$(F(f) : FA \to FB)_{f:A\to B}$$

is bijective.

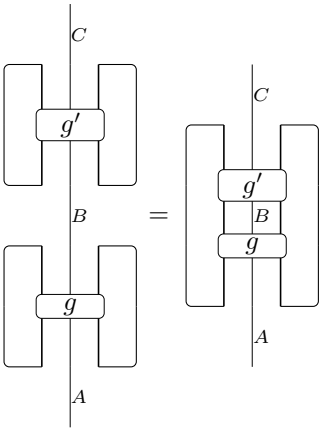
In other words, there is a functional box such that



One can make the box better-looking

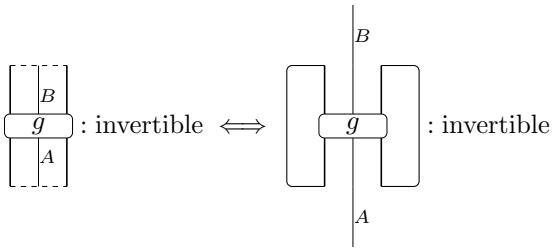


Proposition 4.13 This box has a functoriality-like property:



Combined with proposition 4.3,

Proposition 4.14



5 Natural Transformations

5.1 The Definition

Definition 5.1 (Naturality) Given two infrafunctors

$$F, G : \mathcal{C} \to \mathcal{D}$$

a family of morphisms

$$(\tau_A \in \mathcal{D}(FA, GA))_{A \in \mathrm{Ob}(\mathcal{C})}$$

is called *natural* when for any $f \in \mathcal{C}(A, B)$,

$$\tau_B \circ F(f) = G(f) \circ \tau_A$$

In case parenthese are cumbersome, you can say "τ_A is *natural in* A".

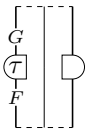
Definition 5.2 (Natural Transformation) Furthermore, in particular case F and G are functorial(then they are functors), τ is denoted as a *natural transformation*

$$\tau : F \to G$$

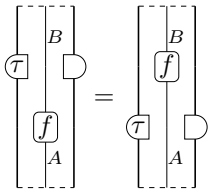
Remark 5.3 The orthogonality of functoriality and naturality is sometimes helpful.

5.2 Natural Connectors

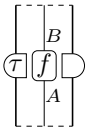
In string diagrams, a natural transformation is a connector of two tubes



because the naturality states a node can travel between tubes:



which inspires you to assign



Definition 5.4 (Vertical Composition) Given three functors

$$F, G, H : \mathcal{C} \to \mathcal{D}$$

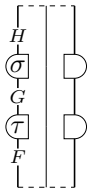
and two natural transformations

$$\tau : F \to G$$
$$\sigma : G \to H$$

the *vertical composition* of τ and σ

$$\sigma \circ \tau : F \to H$$

is defined by



Definition 5.5 (Horizontal Composition) Given four functors

$$F, G : \mathcal{A} \to \mathcal{B}$$
$$H, K : \mathcal{B} \to \mathcal{C}$$

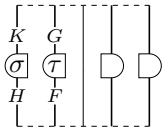
and two natural transformations

$$\tau : F \to G$$
$$\sigma : H \to K$$

the *horizontal composition* of τ and σ

$$\sigma\tau : H \circ F \to K \circ G$$

is defined by



You can easily check the naturality. Travel by car ferry.

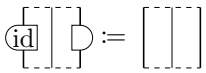
Definition 5.6 (Identity Natural Transformation) Given a functor

$$F : \mathcal{C} \to \mathcal{D}$$

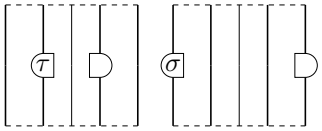
the *identity natural transformation*

$$\mathrm{id}_F : F \to F$$

is defined by



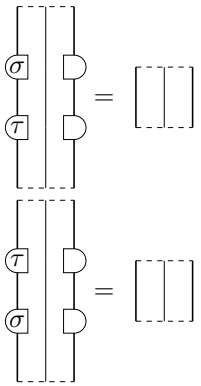
Definition 5.7 (Whiskering) A *whikering* is a horizontal composition with identity natural transformations:



Definition 5.8 (Natural Isomorphism) A *natural isomorphism* is a pair of natural transformations

$$\tau : F \to G$$
$$\sigma : G \to F$$

satisfying the *invertibilty*:



The same symbol is often used for the pair.

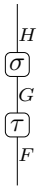
Proposition 5.9 For any natural transformation τ,

$$(\forall A)(\tau_A : \text{invertible})$$

is enough to build the other natural σ.

Definition 5.10 (Functor Category) Given a small category $\mathcal{C}$ and a category $\mathcal{D}$, the functor category $[\mathcal{C}, \mathcal{D}]$ is a category whose objects are functors

from $\mathcal{C}$ to $\mathcal{D}$ and whose morphisms are natural transformations:



, where vertical compositions join the strings.

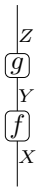
Definition 5.11 For the later use, define a lambda-tasted notation for a set of natural transformations:

$$\mathrm{Nat}_A(FA, GA) \coloneqq \mathrm{Nat}(F, G) \coloneqq [\mathcal{C}, \mathcal{D}](F, G)$$

6 Category of Sets

6.1 The Definition

Definition 6.1 (Category of Sets) The *category of sets* **Set** is a category whose objects are sets and whose morphisms are functions:

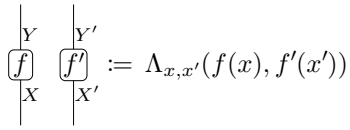


, where nodes are joined by the function composition.

A category is essentially one-dimensional so far: the vertical composition only. Here we introduce the horizontal composition for functions.

Definition 6.2 (Monoidal Category of Sets) Parallel strings are defined by

The horizontal composition is defined by



Strings for the singleton set $\{*\}$ is omitted so that an element of a set is represented as

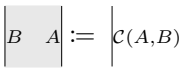
One can check any string diagram built upon these definitions is unambiguous thanks to the trivial bijections:

$$X \times (X' \times X'') \cong (X \times X') \times X''$$
$$X \times \{*\} \cong X$$

Informally such two-dimensional diagrams are called *monoidal*.

6.2 Hom-set Bands

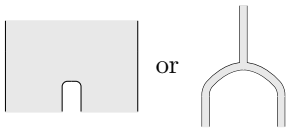
Given a category $\mathcal{C}$, a special string, a *band*, is introduced for hom-sets:



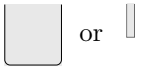
A space-saving form is depicted as

Remark 6.3 Note that the order of objects is flipped. This is resulting from the unfortunate convention that we write $b = h(a)$ but not $h : B \leftarrow A$.

The composition of morphisms can be depicted as



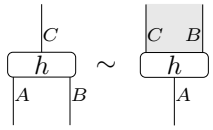
Identity morphisms can be depicted as



As an exercise, write down the associativity and unitality using these diagrams.

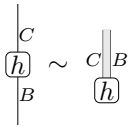
Definition 6.4 (Currying) In particular case $\mathcal{C} = \mathbf{Set}$, there exists the *curry bijection*

$$\mathbf{Set}(A \times B, C) \cong \mathbf{Set}(A, \mathbf{Set}(B, C))$$



We don't distinguish the two diagrams, for the naturality of the bijection ensures "Move the right-side leg up and down" works correct.

Definition 6.5 (Naming) In case A is the singleton set,



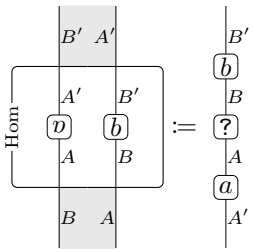
is called a *naming*, which turns a function to an element of function-sets.

Definition 6.6 (Hom-Functor) Hom-sets can be extended to a binary functor

$$\Lambda_{A,B}\mathcal{C}(A, B) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set} \text{ or briefly}$$
$$\mathcal{C}(-, +) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set} \text{ or briefly}$$

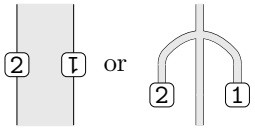
$$\mathrm{Hom}_{\mathcal{C}} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$$

defined by



Remark 6.7 The world in the box is the product category $\mathcal{C}^{\mathrm{op}} \times \mathcal{C}$.

This definition will inspire you to depict the hom-functors as

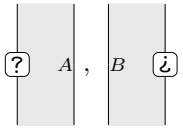


which looks topologically equivalent.

Definition 6.8 (Unary Hom-Functor) According to definition 4.9,

$$\mathcal{C}(A, \texttt{+}) : \mathcal{C} \to \mathbf{Set}$$
$$\mathcal{C}(\texttt{-}, B) : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$$

are respectively depicted as

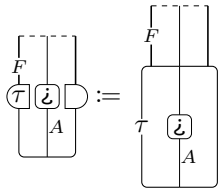


7 The Yoneda Lemma

Definition 7.1 Given a functor $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ and an object A in $\mathcal{C}$, a natural tranformation of the form

$$(\tau_X : \mathcal{C}(X, A) \to FX)_X$$

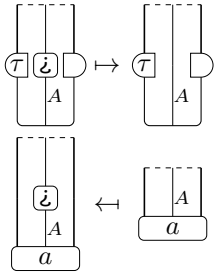
can be depicted as



owing to the naturality.

Definition 7.2 (Yoneda Bijection) The *Yoneda bijection* is defined as

$$\mathrm{Nat}_X(\mathcal{C}(X, A), FX) \cong FA$$

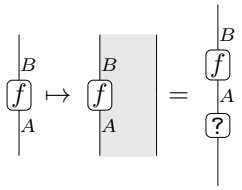


Lemma 7.3 (Yoneda Lemma) The Yoneda bijection is actually bijective and natural in F and A.

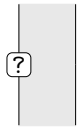
Proof. Now the proof is on my soul trivial! ⌑

Definition 7.4 (Yoneda Embedding) The *Yoneda embedding* is defined as

$$\Lambda_A \Lambda_X \mathcal{C}(X, A) : \mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$$



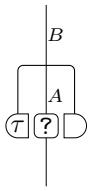
using the diagram of hom functors. In short,



Definition 7.5 A natural transformation of the form

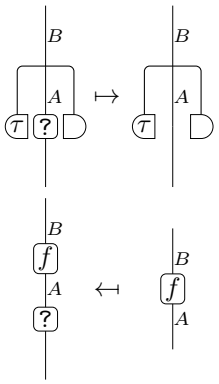
$$(\tau_X : \mathcal{C}(X, A) \to \mathcal{C}(X, B))_X$$

can be depicted as



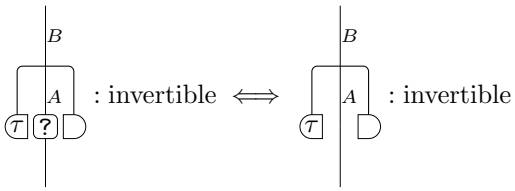
Definition 7.6 (Yoneda Embedding Bijection) In special case $F \coloneqq \mathcal{C}(\text{-}, B)$, the Yoneda bijection is expanded to

$$\mathrm{Nat}_X(\mathcal{C}(X, A), \mathcal{C}(X, B)) \cong \mathcal{C}(A, B)$$



You will notice the second mapping is the Yoneda embedding so that it is full and faithful. Combined with proposition 4.14,

Proposition 7.7 (Yoneda Principle)

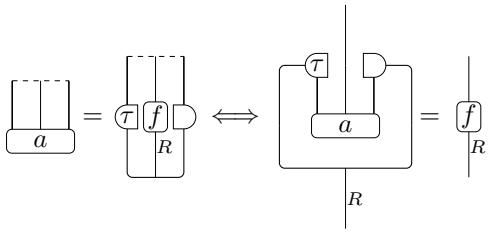


8 Representations

Definition 8.1 (Representation) Given a functor $H : \mathcal{C} \to \mathbf{Set}$, a *representation* of H is a pair of

1. an object R in $\mathcal{C}$
2. a natural bijection $(\tau_X : HX \cong \mathcal{C}(R, X))_X$

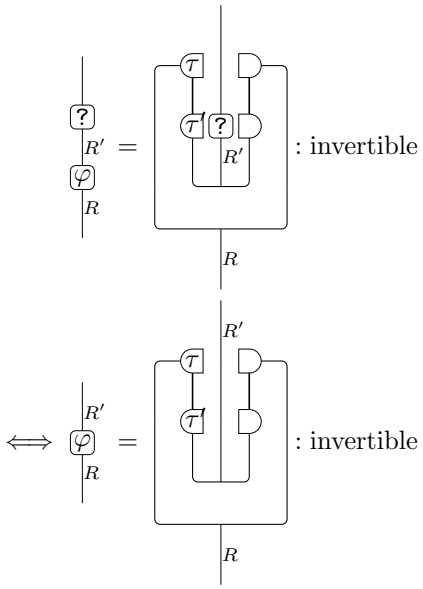
The bijectivity can be expressed using weird boxes



thanks to the naturality. The following proposition allows you to denote the represention of H as rep$H$.

Proposition 8.2 (Uniqueness of Representations) Representations are unique up to unique isomorphism.

Proof. Let (R', τ') be another representation. By the variant of proposition 7.7,

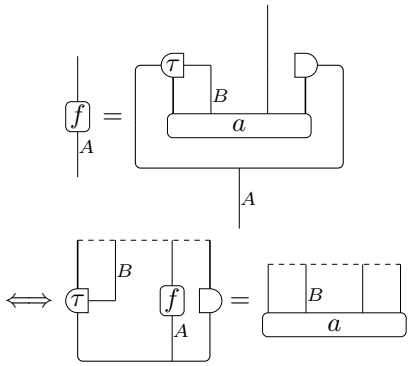


□

Definition 8.3 Given a functor $H : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$, a natural bijection of the form

$$(\tau_X : H(B, X) \cong \mathcal{A}(A, X))_X$$

can be expressed by



Proposition 8.4 (Parameterized Representations) Let $H : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$ be a functor. Given a family of objects $(SB)_B$ and a family of representations

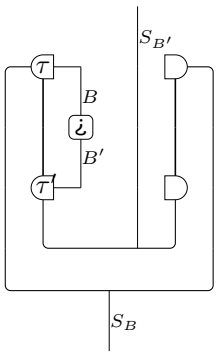
$$((\tau_X^B : H(B, X) \cong \mathcal{A}(SB, X))_X)_B$$

there exists a unique family

$$(S(f) \in \mathcal{A}(SB, SB'))_{f \in \mathcal{B}(B,B')}$$

such that τ is natural in B. Furthermore, S is functorial.

Proof. Define S as



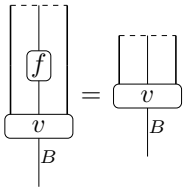
⌗

9 Limits

Definition 9.1 (Cone) Given a functor $F : \mathcal{A} \to \mathcal{B}$, a cone of F consists of

1. an object B in $\mathcal{B}$
2. a natural transformation $(v_X : B \to FX)_X$

Definition 9.2 (Conicality) We may call the naturality of a cone explicitly the *conicality*, which can be expressed as



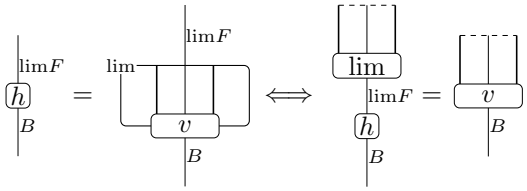
like a magical box any morphism can appear from.

Remark 9.3 Vertical and horizontal composition preserve conicality, a special case of naturality.

Definition 9.4 (Limit) Given a functor $F : \mathcal{A} \to \mathcal{B}$, a limit of F is a pair of

1. an object in $\mathcal{B}$ denoted as $\lim F$
2. a natural bijection $(\mathcal{B}(B, \lim F) \cong \mathrm{Nat}_X(B, FX))_B$

Definition 9.5 (Limiting Cone) The limit bijectivity, thanks to its naturality, can be expressed as



where $\boxed{\text{lim}}$ is a cone called the *limiting cone* of F.

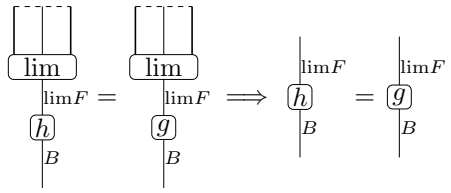
Proposition 9.6 Limits are unique up to isomorphism.

Proof. Immediate by proposition 8.2, because a limit is nothing but a contravariant representation

$$\mathrm{rep}_B \, \mathrm{Nat}_X(B, FX)$$

⌑

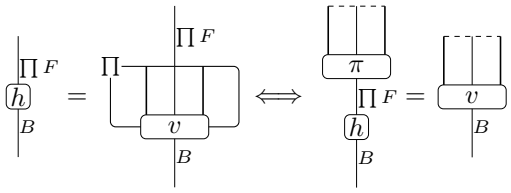
Proposition 9.7 A limiting cone is *monic* meaning that



PROOF. Immediate by the limit bijectivity. ⌑

Definition 9.8 (Product) In particular case the domain of a functor $F : \mathcal{A} \to \mathcal{B}$ is discrete, the limit of F is called the *product* of F denoted as $\prod F$.

Definition 9.9 (Projection) Spelling out the product bijectivity,



where $\boxed{\pi}$ is called the *projection* of F.

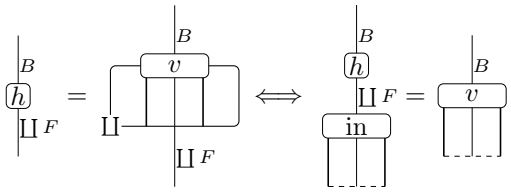
Remark 9.10 Conicality has no concern here, because any family of the form

$$(v_X : B \to FX)_{X \in \mathrm{Ob}(\mathcal{A})}$$

is always natural in case $\mathcal{A}$ is discrete.

Definition 9.11 (Dual) Given a statement containing string diagrams, by flipping it upside down, a corresponding statement is obtained. It is called the *dual* of the original one.

Definition 9.12 (Coproduct) A *coproduct* is a structure obtained from the bijectivity diagram of products flipped.



Remark 9.13 The dual makes a codomain opposite, while the variant does for a domain.

Definition 9.14 (Preservation of Limits) Given a functor $F : \mathcal{A} \to \mathcal{B}$ and a limiting cone of F

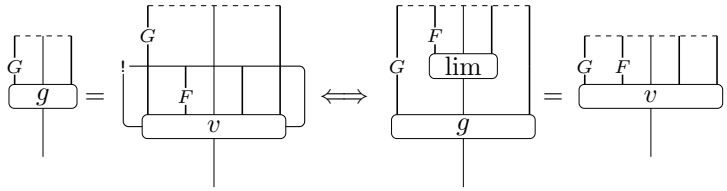
$$(\lim_X : \lim F \to FX)_X$$

a functor $G : \mathcal{B} \to \mathcal{C}$ *preserves limits* of F when

$$(G(\lim_X) : G\lim F \to GFX)_X$$

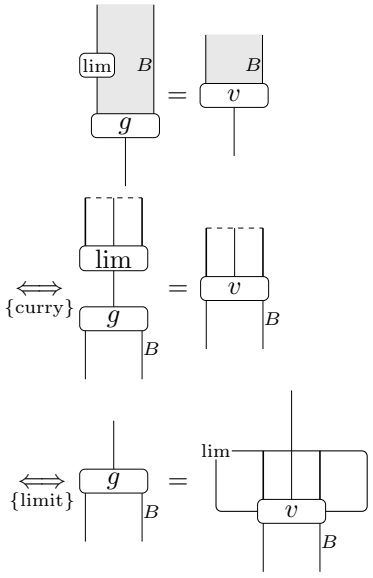
is a limiting cone of $G \circ F$.

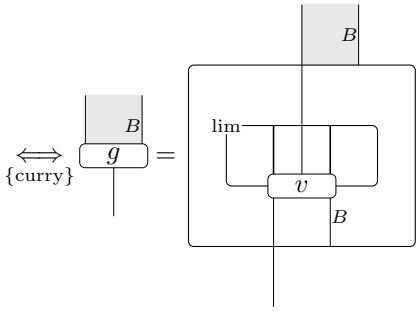
In diagrams, G is such that there exists some box ! satisfying



Proposition 9.15 (HFPL) Hom-functors preserve limits, meaning that given a functor $F : \mathcal{A} \to \mathcal{B}$ and an object B in $\mathcal{B}$, the covariant hom-functor $\mathcal{B}(B, +) : \mathcal{B} \to \mathbf{Set}$ preserves limits of F.

Proof.





$\square$

10 Adjunctions

Definition 10.1 (Adjunction) Given two categories $\mathcal{C}$ and $\mathcal{D}$, an *adjunction*

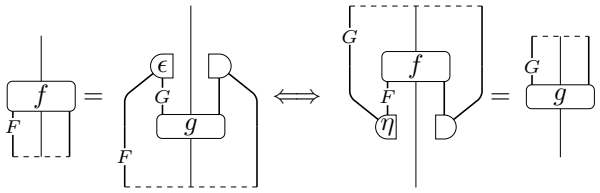
$$F \dashv G$$

consists of

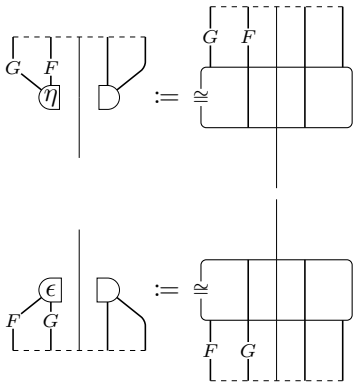
1. *left adjoint*: a functor $F : \mathcal{C} \to \mathcal{D}$
2. *right adjoint*: a functor $G : \mathcal{D} \to \mathcal{C}$
3. *adjunct*: a natural bijection

$$(\mathcal{D}(FC, D) \cong \mathcal{C}(C, GD))_{C,D}$$

A nice consequence is that this bijectivity needs no boxes, expressed by natural transformations only.



where



called respectively the *unit* and *counit*.

Proposition 10.2 Given a functor $G : \mathcal{D} \to \mathcal{C}$, a family of natural bijections

$$((\mathcal{C}(C, GD) \cong \mathcal{D}(F_c, D))_D)_C$$

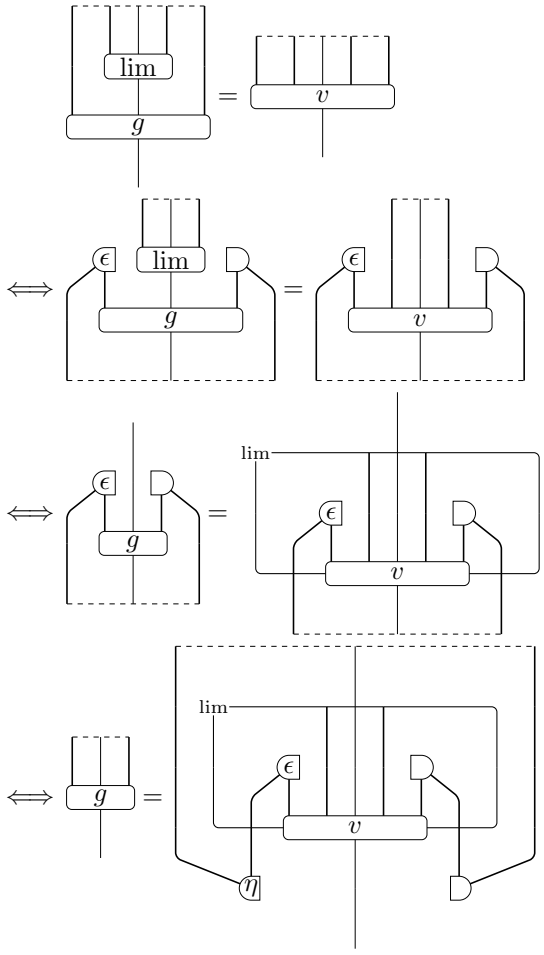
is enough to construct the adjunction $F \dashv G$.

Proof. Immediate by $H(C, D) \coloneqq \mathcal{C}(C, GD)$ in proposition 8.4. ⌑

Proposition 10.3 (RAPL) Right adjoints preserve limits, meaning that given an adjunction $F \dashv (G : \mathcal{D} \to \mathcal{C})$ and a functor $T : \mathcal{B} \to \mathcal{D}$,

$$\begin{aligned} &(\lim_X : \lim T \to TX)_X : \text{limiting cone} \\ \implies &(G(\lim_X) : G\lim T \to GTX)_X : \text{limiting cone} \end{aligned}$$

Proof.



□

11 Monads

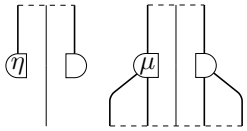
Definition 11.1 (Monad) Given a category $\mathcal{C}$, a monad consists of

1. a functor $T : \mathcal{C} \to \mathcal{C}$
2. *unit*: a natural transformation $\eta : \mathrm{Id}_T \to T$
3. *multiplication*: a natural transformation $\mu : T \circ T \to T$

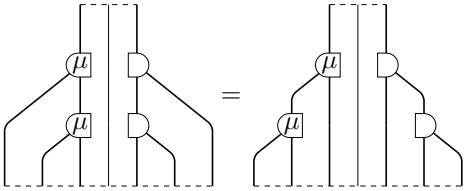
satisfying the coherence conditions

1. *associativity*: $\mu \circ T\mu = \mu \circ \mu T$
2. *unitality*: $\mu \circ T\eta = \mathrm{Id}_T = \mu \circ \eta T$

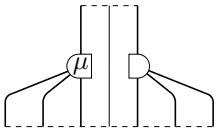
A unit and multiplication are depicted respectively as



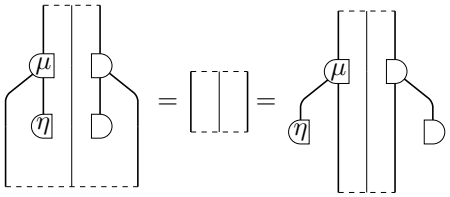
The associativity is depicted as



which inspires you to assign



The unitality is



References

[1] Jiri Adamek, Horst Herrlich, and George E Strecker. *Abstract and Concrete Categories: The Joy of Cats (Dover Books on Mathematics)*. Dover Publications, 8 2009.

[2] Steve Awodey. *Category Theory (Oxford Logic Guides)*. Oxford University Press, 2 edition, 8 2010.

[3] Francis Borceux. *Handbook of Categorical Algebra: Volume 1, Basic Category Theory (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 1 edition, 4 2008.

[4] John Bourke and Micah Blake McCurdy. Frobenius morphisms of bicategories, 2009.

[5] Ralf Hinze. Kan extensions for program optimisation or: Art and dan explain an old trick. In *Mathematics of Program Construction*, pages 324–362. Springer, 2012.

[6] Max Kelly. *Basic Concepts of Enriched Category Theory (London Mathematical Society Lecture Note Series)*. Cambridge University Press, 4 1982.

[7] Aleks Kissinger. Pictures of processes: automated graph rewriting for monoidal categories and applications to quantum computing. *arXiv preprint arXiv:1203.0202*, 2012.

[8] Saunders Mac Lane. *Categories for the Working Mathematician (Graduate Texts in Mathematics)*. Springer, 2nd ed. 1978. softcover reprint of the original 2nd ed. 1978 edition, 11 2010.

[9] Dan Marsden. Category theory using string diagrams. *CoRR*, abs/1401.7220, 2014.

[10] Micah Blake McCurdy. Strings and stripes, graphical calculus for monoidal functors and monads, 2010.

[11] Paul-André Mellies. Functorial boxes in string diagrams. In *Computer science logic*, pages 1–30. Springer, 2006.

[12] Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.

[13] Daniele Turi. Category theory lecture notes. *Laboratory for Foundations of Computer Science, University of Edinburgh*, 2001.