Category Theory with Strings

Shunsuke Sogame

March 21, 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

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

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,

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

means "there exists a unique a such that P, furthermore 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. ($\Longrightarrow$) by the Axiom of choice. ($\Longleftarrow$) immediate. $\square$

Definition 2.6 (Functional 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

$$x$$

Definition 2.10 (Family) A *family declaration* is a way to provide a function with arbitary application syntax. The usage is clear from an example

$$(x \in Y)_{x \in X}$$

We call such a function a *family*. Furthermore, a function body can be placed like that

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

Example 2.11 The most-used family declaration is the subscript style $(a_i)_i$. You can view a tuple $(a_1, a_2, \ldots, a_n)$ to be an abbreviation of $(a_i)_{i \in \{1,2,\ldots,n\}}$.

Families 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

It is sure you 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*.

11 Monads

11.1 The Definition

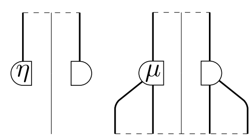
Definition 11.1 (Monad) Given a category $\mathcal{C}$, a *monad* on $\mathcal{C}$ consists of

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 $\mu : T \circ T \to T$

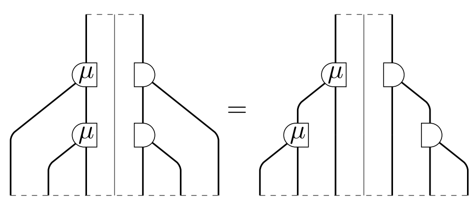
satisfying the coherence conditions

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

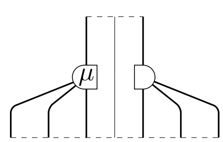
A unit and multiplication are depicted respectively as



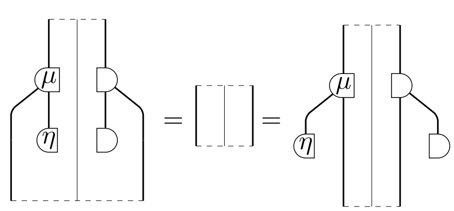
The associativity is depicted as



This inspires you to assign



The unitality is



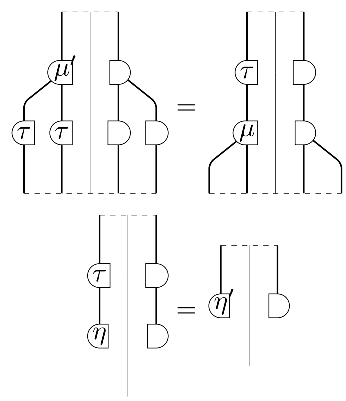
Definition 11.2 (Monad Morphism) Given a category $\mathcal{C}$, a *monad morphism* consists of

1. *domain*: a monad (T, η, μ) on $\mathcal{C}$
2. *codomain*: a monad (T', η', μ') on $\mathcal{C}$
3. a natural transformation $\tau : T \to T'$

satisfying the coherence conditions

1. *multiplication-compatibility*: $\tau \circ \eta = \eta'$
2. *unit-compatibility*: $\tau \circ \mu = \mu' \circ \tau\tau$

The coherence is depicated as

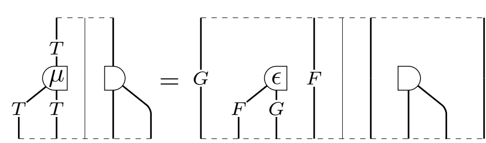


Definition 11.3 (Categoy of Monads) Given a category $\mathcal{C}$, the *category of monads* $\mathbf{Mnd}(\mathcal{C})$ is a category whose objects are monads and whose morphisms are monad morphisms.

Definition 11.4 (Monad-Associated Adjunction) Given a monad (T, η, μ), we call an adjunction $F \dashv G$ T*-associated* when

1. $T = G \circ F$
2. $\mu = G\epsilon F$

This condition can be depicted as

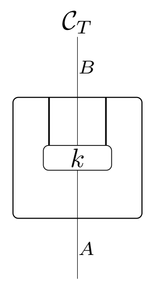


11.2 Kleisli Categories

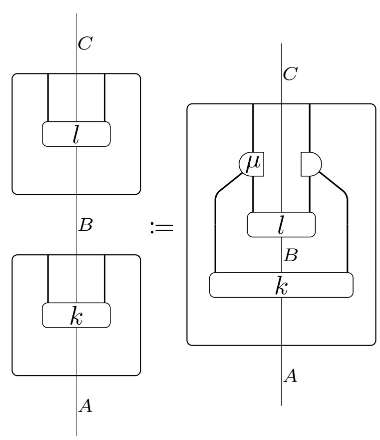
Definition 11.5 (Kleisli Category) Given a monad (T, η, μ) on $\mathcal{C}$, the *Kleisli category* of T, denoted as $\mathcal{C}_T$ is a category consisting of

1. $\mathrm{Ob}(\mathcal{C}_T) \coloneqq \mathrm{Ob}(\mathcal{C})$
2. $\mathcal{C}_T(A, B) \coloneqq \mathcal{C}(A, TB)$
3. $l \circ k \coloneqq \mu \circ T(l) \circ k$
4. $\mathrm{id}_A \coloneqq \eta_A$

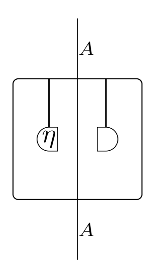
In diagrams, a morphism in $\mathcal{C}_T$ is depicted as a *Kleisli box*



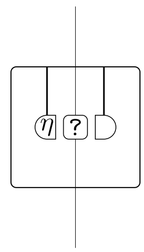
The composition is defined as



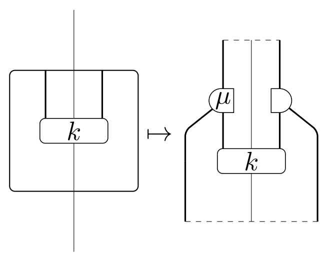
An identity morphism is defined as



Definition 11.6 (Kleisli Adjunction) Define a functor $L : \mathcal{C} \to \mathcal{C}_T$ as



$K : \mathcal{C}_T \to \mathcal{C}$ as



then they consitute the *Kleisli adjunction* $L \dashv K$ whose adjunct is the Kleisli boxing. This adjunction is T-associated.

11.3 Eilenberg-Moore Categories

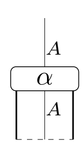
Definition 11.7 (Monad Algebra) Given a monad (T, η, μ) on $\mathcal{C}$, a *monad algebra*, denoted as T-algebra, consists of

1. an object $A \in \mathcal{C}$
2. a morphism $\alpha : TA \to A$

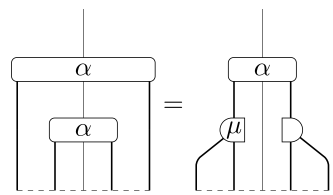
satisfying the coherence

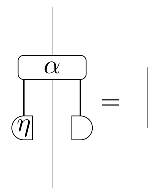
1. *associativity*: $\alpha \circ T(\alpha) = \alpha \circ \mu$
2. *unitialit*: $\alpha \circ \eta = \mathrm{id}$

A T-algebra is depicated as



The coherence can be depicted as





TODO