Notes on Topos Theory and Type Theory

Utensil Song

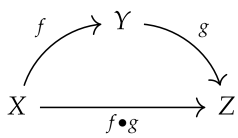
September 2, 2025

1 Category theory

1.1 Categories

Definition 1.1.1 (Category [Kostecki(2011), 1.1])
A **category** C consists of:

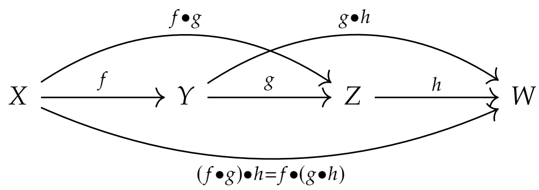
1. **objects** $\mathrm{Ob}(C)$: $O, X, Y, \ldots$
2. **arrows** $\mathrm{Arr}(C)$: $f, g, h, \ldots$, where for each arrow f,
   - a pair of operations dom and cod assign a **domain** object $X = \mathrm{dom}(f)$ and a **codomain** object $Y = \mathrm{cod}(f)$ to f,
   - thus f can be denoted by
   $$f : X \to Y \tag{1.1.2}$$
   or
   $$X \xrightarrow{f} Y \tag{1.1.3}$$
3. **compositions**: a composite arrow of any pair of arrows f and g, denoted $g \circ f$ or $f \bullet g$, makes the diagram

   commute (we say that $f \bullet g$ **factors through** Y),
4. a **identity arrow** for each object O, denoted $1_O : O \to O$

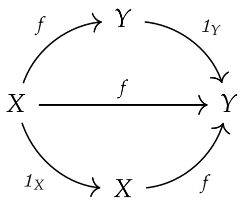
satisfying:

1. **associativity** of composition: the diagram

   commutes,

2. **identity law**: the diagram

   commutes.

Convention 1.1.4 (Composition [Kostecki(2011), 1.1])

In literatures, **composition** of arrows in a **category** is most frequently denoted $\circ$ or just by juxtaposition, i.e. for the diagram

$$\bullet \xrightarrow{f} \bullet \xrightarrow{g} \bullet$$

$g \circ f$ or just gf mean g applies to the result of f.

But this is the opposite of the order of arrows in the diagram, so arguably a more natural way could be to denote the above as $f ; g$ [Kostecki(2011), 1.1], or as $f \ \ g$ [Fong and Spivak(2018), 3.6], or as $f \bullet g$ [Nakahira(2023), sec. 1.1].

We use $f \bullet g$ throughout this note, so it can always be understood as $\xrightarrow{f}\xrightarrow{g}$. This way saves mental energy when reading **commuting diagrams**, **pasting diagrams** and **string diagrams** which we employ heavily.

Definition 1.1.5 ((locally) small, hom-set [Kostecki(2011), 1.3])

A category C is called **locally small** iff for *any* of its objects X, Y, the collection of arrows from X to Y is a **set**, called a **hom-set**, denoted

$$\mathrm{Hom}_C(X,Y) \tag{1.1.6}$$

A category is called **small** iff the collection of *all* its arrows is a **set**.

Notation 1.1.7 (Hom-set, hom-class [Zhang(2021), 1.2])

Hom in $\mathrm{Hom}_C(X,Y)$ is short for homomorphism, since an arrow in category theory is a morphism (i.e. an arrow), a generalization of homomorphism between algebraic structures.

This notation could be unnecessarily verbose, so when there is no confusion, we follow [Leinster(2016)] and [Zhang(2021)] to simply write $X,Y \in \mathrm{Ob}(C)$ as

$$X,Y \in C \tag{1.1.8}$$

and $f \in \mathrm{Hom}_C(X,Y)$ as

$$f \in C(X,Y) \tag{1.1.9}$$

In some other scenarios, when the category in question is clear (and it might be to too verbose to write out, e.g. a **functor category**), we omit the subscript of the category and write just

$$\mathrm{Hom}(X,Y) \tag{1.1.10}$$

In general, collection $\mathrm{Ob}(C)$ and $\mathrm{Arr}(C)$ are not neccessarily sets, but classes. In that case, $\mathrm{Hom}_C(X,Y)$ is called a **hom-class**.

Later, we will also learn that Ob and Arr are representable functors.

Definition 1.1.11 (Finite [Kostecki(2011), 1.3])

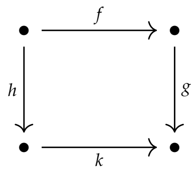
A category is called **finite** iff it is **small** and it has only a finite number of objects and arrows.

Definition 1.1.12 (Commuting diagram [Kostecki(2011), 1.2])

A **diagram** in a category C is defined as a collection of objects and arrows that belong to the category C with specified operations dom and cod.

A **commuting diagram** is defined as such diagram that any two arrows of this diagram which have the same domain and codomain are equal.

For example, that the diagram



commutes means $f \bullet g = h \bullet k$.

It's also called a **arrow diagram** [Nakahira(2023), 1.1] when compared to a **string diagram**, as it represents **arrow**s with $\rightarrow$.

Definition 1.1.13 (String diagram [Nakahira(2023), table 1.1])

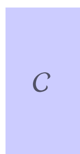
A **string diagram** represents categories as surfaces (2-dimensional), functors as wires (1-dimensional), natural transformations as blocks or just dots (0-dimensional).

String diagrams has the advantage of being able to represent objects, arrows, functors, and natural transformations from multiple categories, and their vertical and horizontal composition, and has various **topologically plausible** calculational rules for proofs.

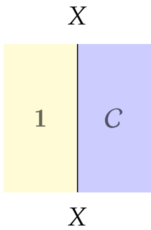
Notation 1.1.14 (String diagrams: category, object and arrow [Marsden(2014), sec. 2.1])

Later, when we have learned about functors and natural transformations, we will see that, in **string diagram** for 1-category:

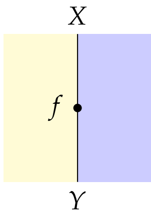
1. A **category** C is represented as a colored region:



2. **Functor**s of type $\mathbf{1} \rightarrow C$ can be identified with objects of the category C, where $\mathbf{1}$ is the the **terminal category**, so an **object** $X \in \mathrm{Ob}(C)$ can be represented as:



3. An **arrow** $f : X \to Y$ is then a **natural transformation** between two of these functors, represented as:



Convention 1.1.15 (Letters)

The general idea is that we try to use visually distinct letters for different concepts:

1. **uppercase calligraphic letter**s $\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{J}, \mathcal{S}$ denote **categories**
   - $\mathcal{C}, \mathcal{D}, \mathcal{E}$ are prefered in concepts about one to three categories, since $\mathcal{C}$ is the first letter of "category"
   - $\mathcal{J}$ is used in concepts like **diagram**, and assumed to be **small**
   - $\mathcal{S}$ is used in concepts like **subcategory**, or sometimes in concepts about a **small** category
2. **boldface uppercase Roman letter**s denote specific categories, e.g. **Cat**, **Set**, **Grp**, **Top**, **1**
3. **uppercase Roman letter**s X, Y, Z, W, O, E, V denote **object**s in categories
   - X, Y usually mean objects in $\mathcal{C}$ and $\mathcal{D}$, respectively
   - O denotes any object
   - E denotes the **equalizer** object
   - in **limit**-related concepts, $-$ denotes any object, V denotes the vertex
4. **lowercase Roman letter**s f, g, h, i, k, l, r and sometimes the lowercase Roman letter of the corresponding codomain or domain object denote **arrow**s
   - occationally, when two arrows are closely related, they are denoted by the same letter with different subscripts, e.g. g_1, g_2

- as special cases, ι, p, i denote the inclusion, projection and injection arrows

5. **uppercase script letter**s $\mathscr{F}, \mathscr{G}, \mathscr{H}, \mathscr{I}, \mathscr{K}, \mathscr{L}, \mathscr{R}$ denote **functor**s
   - $\mathscr{D}$ denotes a **diagram** functor
   - $\mathscr{L}$ and $\mathscr{R}$ denote the left and right adjoint functors in an **adjunction**, respectively
   - $\mathscr{H}$ denotes the **Yoneda embedding functor**s
   - as a special case, functors with the **terminal category** (i.e. **constant object functor**s) as the domain are identified with the objects in the codomain category, thus are denoted like an object: $X : \mathbf{1} \to C, * \mapsto X$
   - $\mathscr{I}$ is only used in the **inclusion functor** (note that this is letter "I")
   - we do not use $\mathscr{J}$ and $\mathscr{S}$ because they are visually ambiguous
6. **lowercase Greek letter**s $\alpha, \beta, \eta, \epsilon, \sigma$ denote **natural transformation**s, their components are denoted by them with subscripts.

1.2 Isomorphism

Definition 1.2.1 (Monic [Kostecki(2011), 2.1])
An arrow $f : X \to Y$ is **monic** if the diagram

$$O \underset{g_2}{\overset{g_1}{\rightrightarrows}} X \xrightarrow{f} Y$$

commutes, i.e. $g_1 \bullet f = g_2 \bullet f \implies g_1 = g_2$, denoted $f : X \rightarrowtail Y$.

"Monic" is short for "monomorphism", which is a generalization of the concept of injective (one-to-one) functions between sets.

Definition 1.2.2 (Epic [Kostecki(2011), 2.2])
An arrow $f : X \to Y$ is **epic** if the diagram

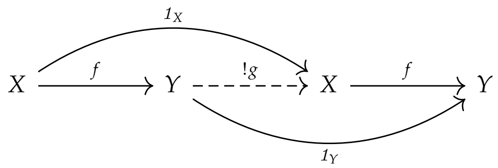
$$X \xrightarrow{f} Y \underset{g_2}{\overset{g_1}{\rightrightarrows}} O$$

commutes, i.e. $f \bullet g_1 = f \bullet g_2 \implies g_1 = g_2$, denoted $f : X \twoheadrightarrow Y$.

"Epic" is short for "epimorphism", which is a generalization of the concept of surjective (onto) functions between sets.

Definition 1.2.3 (Iso [Kostecki(2011), 2.3])

An arrow $f : X \to Y$ is **iso**, or X and Y are **isomorphic**, denoted $X \cong Y$, or $X \xrightarrow{\sim} Y$, if the diagram

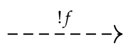


commutes, where $!g$ means there exists a unique arrow g, and g is called the **inverse** of f, denoted f^{-1}.

"Iso" is short for "isomorphism", which is a generalization of the concept of bijective (one-to-one and onto) functions.

Convention 1.2.4 (Uniqueness: dashed arrow)

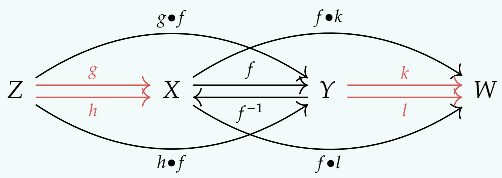
Uniqueness of an arrow is denoted $\exists! f$ or simply $!f$, and visualized as a **dashed arrow** in diagrams, and ! is often omitted.



Lemma 1.2.5 (Iso [Kostecki(2011), 2.4])

An iso arrow is always monic and epic. However, not every arrow which is monic and epic is also iso.

Proof. The diagram



commutes for the iso arrow f, thus

- $g = h$ i.e. f is monic,
- $k = l$ i.e. f is epic.

□

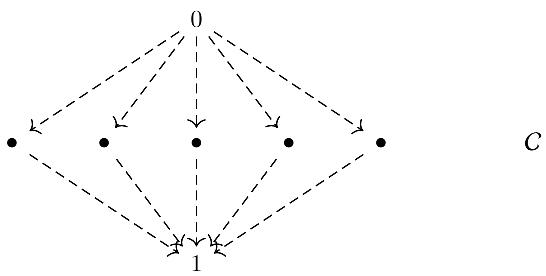
1.3 Special objects and categories

Definition 1.3.1 (Initial, terminal and null objects [Kostecki(2011), 2.6])

An **initial object** in a category C is an object $0 \in \mathrm{Ob}(C)$ such that for any object X in C, there exists a unique arrow $0 \to X$. It's also called a **universal object**, or a **free object**.

A **terminal object** in a category C is an object $1 \in \mathrm{Ob}(C)$ such that for any object X in C, there exists a unique arrow $X \to 1$. It's also called a **final object**, or a **bound object**.

Diagramatically,



A **null object** is an object which is both **terminal** and **initial**, confusingly, it's also called a **zero object**.

Lemma 1.3.2 (Uniqueness [Kostecki(2011), 2.7])

All **initial objects** in a category are isomorphic.

All **terminal objects** in a category are isomorphic.

In other words, they are unique up to isomorphism, respectively.

Definition 1.3.3 (Element [Kostecki(2011), 2.8, 2.9])

Let $X, S \in \mathrm{Ob}(C)$.

An **element** or a **generalized element** of X at **stage** S (or, of **shape** S) is an arrow $x : S \to X$ in C, also denoted $x \in_S X$.

An arrow $1 \to X$ is called a **global element** of X, a.k.a. a **point** of X.

An arrow $S \to X$, if S is not isomorphic to 1, is called the **local element** of X at stage S.

An arrow $1_X : X \to X$ is called the **generic element** of X.

Remark 1.3.4 (Element [Kostecki(2011), 2.8])

In an **element** $x : S \to X$, the object S is called a **stage** in order to express the intuition that it is a "place of view" on X. In the same sense, S is also called a **domain of variation**, and X a **variable element**.

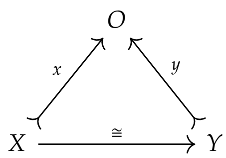
Sometimes, the term **shape** is used instead [Leinster(2016), 4.1.25], intuitive examples are:

- when the object is a set, a generalized element of X of shape $\mathbb{N}$ is a sequence in the set X
- when the object is a topological space, a generalized element of X of shape S^1 is a loop

In the context of studying solutions to polynomial equations, we may also call it a S-valued **point** in X, where S is the number set where the solution is taken, e.g. the real, complex, and $\operatorname{Spec} \mathbb{F}_p$-valued solutions.

Definition 1.3.5 (Equivalent, equivalence class [Kostecki(2011), 2.10])

Two monic arrows x and y which satisfy



are called **equivalent**, which is denoted as $x \sim y$.

The **equivalence class** of x is denoted as $[x]$, i.e., $[x] = \{y \mid x \sim y\}$.

Definition 1.3.6 (Subobject, Sub [Kostecki(2011), 2.10])

A **subobject** of any object is defined as an **equivalence class** of monic arrows into it.

The **class of subobjects** of an object X is denoted as

$$\operatorname{Sub}(X) := \{[f] \mid \operatorname{cod}(f) = X \wedge f \text{ is monic}\}. \tag{1.3.7}$$

Definition 1.3.8 (Set [Kostecki(2011), 1.1, example 1])

Set, the **category** of sets, consists of objects which are sets, and arrows which are functions between them. The axioms of composition, associativity and identity hold due to standard properties of sets and functions.

Set has the **initial object** $\varnothing$, the empty set, and the **terminal object**, $\{*\}$, the singleton set.

Set doesn't have a **null object**.

Monic arrows in **Set** are denoted by $f : X \hookrightarrow Y$, interpreted as an inclusion map (see also inclusion function in nLab).

Given X : **Set**, the **subobject**s of X are in canonical one-to-one correspondence with the **subset**s of X.

Notation 1.3.9 (Inclusion [Leinster(2016), 0.8])

In

$$X \xhookrightarrow{\iota} Y$$

the symbol $\hookrightarrow$ is used for inclusions. It is a combination of a subset symbol $\subset$ and an arrow.

Definition 1.3.10 (Cat [Leinster(2016), 3.2.10])

We denote by **Cat** the category of **small** categories and functors between them.

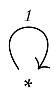
Definition 1.3.11 (Discrete category [Leinster(2016), 1.1.8])

A **discrete category** has no arrows apart from the identity arrow, i.e. it amounts to just a class of objects.

We can regard a set as a discrete category.

Definition 1.3.12 (Terminal category [Leinster(2016), 4.1.6])

A **terminal category**, denoted **1**, has only one object, denoted $*$, and only the identity arrow, denoted $\mathit{1}$.



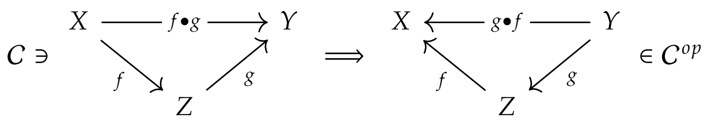
1

Definition 1.3.13 (Opposite category [Kostecki(2011), 2.5])
A category is called the **opposite category** of $\mathcal{C}$, denoted $\mathcal{C}^{op}$, iff

1. (reversion of arrows)
$$\mathrm{Ob}(\mathcal{C}^{op}) = \mathrm{Ob}(\mathcal{C}) \tag{1.3.14}$$

$$\mathrm{Arr}(\mathcal{C}^{op}) \ni f : Y \to X \iff \mathrm{Arr}(\mathcal{C}) \ni f : X \to Y \tag{1.3.15}$$

2. (reversion of composition)



Definition 1.3.16 (Product category [Leinster(2016), 1.1.11])
Given categories $\mathcal{C}$ and $\mathcal{D}$, there is a **product category**, denoted $\mathcal{C} \times \mathcal{D}$, in which

- an object is a pair (X, Y)
- an arrow $(X, Y) \to (X', Y')$ is a pair (f, g)
- the composition is given by
$$(f_1, g_1) \bullet (f_2, g_2) = (f_1 \bullet f_2, g_1 \bullet g_2) \tag{1.3.17}$$
- the identity on (X, Y), denoted $\mathit{1}_{(X,Y)}$ is $(\mathit{1}_X, \mathit{1}_Y)$

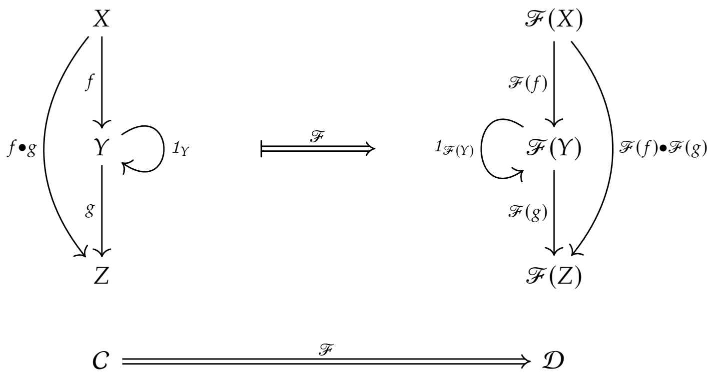
where $X \in \mathcal{C}$, $Y \in \mathcal{D}$, $f : X \to X' \in \mathcal{C}$, and $g : Y \to Y' \in \mathcal{D}$.

Definition 1.3.18 ((full) subcategory [Leinster(2016), 1.2.18])
Let $\mathcal{C}$ be a category. A **subcategory** $\mathcal{S}$ of $\mathcal{C}$ consists of a subclass $\mathrm{Ob}(\mathcal{S})$ of $\mathrm{Ob}(\mathcal{C})$ together with, for each $S, S' \in \mathrm{Ob}(\mathcal{S})$, a subclass $\mathcal{S}(S, S')$ of $\mathcal{C}(S, S')$, such that $\mathcal{S}$ is closed under composition and identities.

It is a **full subcategory** if $\mathcal{S}(S, S') = \mathcal{C}(S, S')$ for all $S, S' \in \mathrm{Ob}(\mathcal{S})$.

1.4 Functors

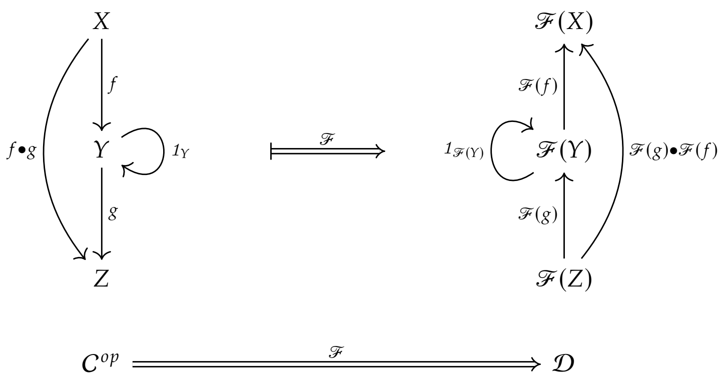
Definition 1.4.1 ((covariant) functor [Kostecki(2011), 3.1])
A **(covariant) functor** $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ is given by the diagram



i.e. a map of objects and arrows between categories $\mathcal{C}$ and $\mathcal{D}$ that *preserves* the structure of the compositions and identities.

Definition 1.4.2 (Contravariant functor [Kostecki(2011), 3.1])

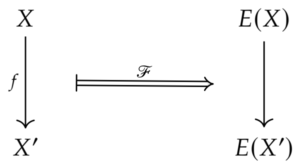
A functor $\mathscr{F}$ is called a **contravariant functor** from $\mathcal{C}$ to $\mathcal{D}$, and denoted $\mathscr{F} : \mathcal{C}^{op} \to \mathcal{D}$, if it obeys the definition given by the **(covariant) functor** for $\mathcal{C}$ replaced by $\mathcal{C}^{op}$, i.e. it's given by the diagram



i.e. a map of objects and arrows between categories $\mathcal{C}$ and $\mathcal{D}$ that *reverses* the structure of the arrows, compositions and identities.

Definition 1.4.3 (Functorial in [Leinster(2016), sec. 4.1])

For some expression $E(X)$ containing X, when we say $E(X)$ is (covariant) **functorial in** X, we mean that there exists a functor $\mathscr{F}$ such that

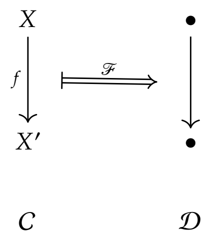


for every $f : X \to X'$.

Dually, we use the term **contravariantly functorial in**.

Convention 1.4.4 (Functors)

For simplicity, when there is no confusion, we use $\bullet$ to represent corresponding objects, and omit the arrow names in the codomain of a functor, e.g.



Definition 1.4.5 (Full and faithful [Kostecki(2011), 3.2])

A functor $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ is **full** iff for any pair of objects X, Y in $\mathcal{C}$ the induced map $F_{X,Y} : \mathcal{C}(X, Y) \to \mathcal{D}(\mathscr{F}(X), \mathscr{F}(Y))$ is surjective (onto). $\mathscr{F}$ is **faithful** if this map is injective (one-to-one).

Definition 1.4.6 (Preserve and reflect [Kostecki(2011), 3.3])

A functor $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ is called to **preserve** a property $\wp$ of an arrow iff for every $f \in \mathrm{Arr}(\mathcal{C})$ that has a property $\wp$ it follows that $\mathscr{F}(f) \in \mathrm{Arr}(\mathcal{D})$ has this property. A functor $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ is called to **reflect** a property $\wp$ of an arrow iff for every $\mathscr{F}(f) \in \mathrm{Arr}(\mathcal{D})$ that has a property $\wp$ it follows that $f \in \mathrm{Arr}(\mathcal{C})$ has this property.

Example 1.4.7 (Full, faithful, preserve and reflect [Kostecki(2011), 3.3])

Every inclusion functor is faithful.

Every functor preserves isomorphisms.

Every faithful functor reflects monomorphisms and epimorphisms.

Every full and faithful functor reflects isomorphisms.

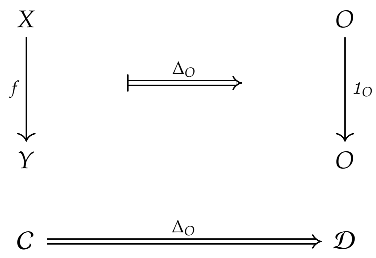
1.5 Special functors

Definition 1.5.1 (Identity functor [Kostecki(2011), 3.1, example 1])

The **identity functor** $1_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$ (denoted also by $1_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$), defined by $1_{\mathcal{C}}(X) = X$ and $1_{\mathcal{C}}(f) = f$ for every $X \in \mathrm{Ob}(\mathcal{C})$ and every $f \in \mathrm{Arr}(\mathcal{C})$.

Definition 1.5.2 (Constant functor [Kostecki(2011), 3.1, example 2])

The **constant functor** $\Delta_O : \mathcal{C} \to \mathcal{D}$ which assigns a fixed $O \in \mathrm{Ob}(\mathcal{D})$ to any object of $\mathcal{C}$ and 1_O, the identity arrow on O, to any arrows from $\mathcal{C}$:



with compositions and identities preserved in a trivial way.

Definition 1.5.3 (Constant object functor [Leinster(2016), 4.1.6])

A **functor** from the **terminal category** $\mathbf{1}$ to a category $\mathcal{C}$ simply picks out an object of $\mathcal{C}$, called a **constant object functor** (which is a **constant functor**), denoted $\Delta_X : \mathbf{1} \to \mathcal{C}$ for some $X \in \mathrm{Ob}(\mathcal{C})$, or simly denoted by the object, e.g. X.

As special cases, constant object functor for initial and terminal objects are denoted by 0 and 1, respectively.

Definition 1.5.4 (Unique functor [Nakahira(2023), eq. 2.3])

A **unique functor**, is a functor from a category $\mathcal{C}$ to the **terminal category** $\mathbf{1}$, uniquely determined by mapping all arrows in $\mathcal{C}$ to the identity arrow 1_* of the unique object $*$ in $\mathbf{1}$.

This functor is often denoted by $! : \mathcal{C} \to \mathbf{1}$.

Intuitively, the functor ! acts to erase all information about the input.

Definition 1.5.5 (Diagonal functor [Leinster(2016), sec. 6.1])

Given a small category $\mathcal{J}$ and a category $\mathcal{C}$, the **diagonal functor**

$$\Delta_{\mathcal{J}} : \mathcal{C} \to [\mathcal{J}, \mathcal{C}] \tag{1.5.6}$$

maps each object $X \in \mathcal{C}$ to the **constant functor** $\Delta_{\mathcal{J}}(X) : \mathcal{J} \to \mathcal{C}$, which in turn maps each object in $\mathcal{J}$ to X, and all arrows in $\mathcal{J}$ to 1_X.

When $\mathcal{J}$ is clear in the context, we may write $\Delta_{\mathcal{J}}(X)$ as Δ_X.

Particularly [Kostecki(2011), 3.1, example 6], when $\mathcal{J}$ is a discrete category of two objects,

$\Delta : \mathcal{C} \to \mathcal{C} \times \mathcal{C}, \Delta(X) = (X, X)$ and $\Delta(f) = (f, f)$ for $f : X \to X'$

$\Delta_{\mathcal{J}}(X)$ is the same as $X\bullet!$, thus [Nakahira(2023), eq. 2.12]

$$\Delta_{\mathcal{J}} = -\bullet! \tag{1.5.7}$$

Definition 1.5.8 (Forgetful functor [Kostecki(2011), 3.1, example 3])

The **forgetful functor**, which *forgets* some part of structure, however arrows, compositions and identities are preserved.

Definition 1.5.9 (Inclusion functor [Leinster(2016), 1.2.18])

Whenever $\mathcal{S}$ is a **subcategory** of a category $\mathcal{C}$, there is an **inclusion functor** $\mathscr{I} : \mathcal{S} \hookrightarrow \mathcal{C}$ defined by $\mathscr{I}(S) = S$ and $\mathscr{I}(f) = f$, i.e. it sends objects and arrows of $\mathcal{S}$ into themselves in category $\mathcal{C}$. It is automatically **faithful**, and it is **full** iff S is a full subcategory.

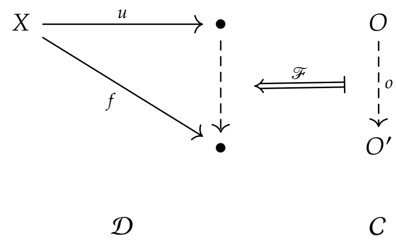
Example 1.5.10 (Other special functors [Kostecki(2011), 3.1, example 10, 11, 4.6])

Some other special functors are introduced in later sections in context, e.g. **hom-functor**, **Yoneda embedding functors**.

1.6 Universal properties

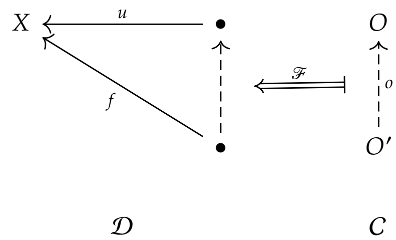
Definition 1.6.1 (Universal arrow [Kostecki(2011), 3.4, 3.5])

A **universal arrow** from $X \in \mathcal{D}$ to $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ is a unique pair (O, u) that makes the diagram



commute for any $f \in \mathcal{D}$.

Conversely, a **(co)universal arrow** from $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ to $X \in \mathcal{D}$ is a unique pair (O, u) that makes the diagram



commute for any $f \in \mathcal{D}$.

Remark 1.6.2 (Universal property [Leinster(2016), sec. 1])

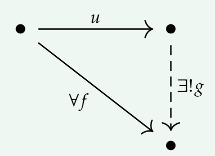
We say **universal arrows** satisfy some **universal property**.

The term **universal property** is used to describe some **property** (usually some equality of some compositions, or a **commuting diagram** in general) that is satisfied *for all* (hence **universal**) relevant objects/arrows in the "world" (i.e. in the relevant categories), by a corresponding *unique* object/arrow.

It's usually spell out like this:

> In a context where (usually a given diagram), for all (some objects/arrows), there exists a unique (object/arrow) such that (some property).

Diagramatically,



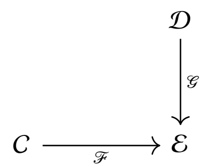
For clarity and brevity, usually the ∀ clauses are specified outside the diagram, and the ∃! clause is expressed by dashed arrows. For more elaborate diagrams, see [Freyd(1976)][Fong and Spivak(2018)][Ochs(2022)].

For the diagram above, we also say that f *uniquely* **factors through** u along g [Riehl(2017), 2.3.7].

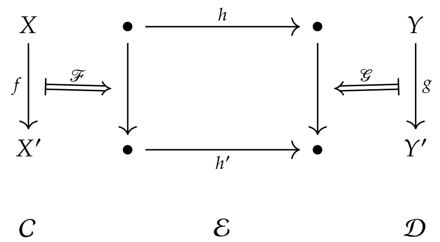
A universal property is a property of some construction which boils down to (is manifestly equivalent to) the property that an associated object is an **initial object** of some (auxiliary) category [nLab(2020)].

Definition 1.6.3 (Comma category [Leinster(2016), 2.3.1])

Given categories and functors



the **comma category** $\mathscr{F} \downarrow \mathscr{G}$ (or $(\mathscr{F} \Rightarrow \mathscr{G})$) is the category given by objects (X, h, Y) and arrows (f, g) that makes the diagram



commute.

Lemma 1.6.4 (Universal arrow via initial/terminal object of comma category [Kostecki(2011), 3.6])
A universal arrow from $X \in \mathcal{D}$ to $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ is an initial object in the comma category $X \downarrow \mathcal{F}$.

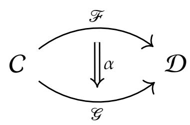
Conversely, a (co)universal arrow from $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ to $X \in \mathcal{D}$ is a terminal object in the comma category $\mathcal{F} \downarrow X$.

1.7 Natural transformation and functor category

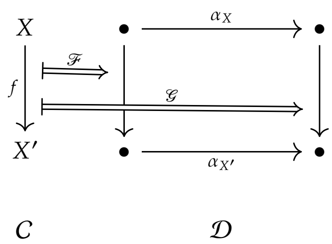
Definition 1.7.1 (Natural transformation [Leinster(2016), 1.3.1])
Given categories and functors

$$\mathcal{C} \underset{\mathcal{G}}{\overset{\mathcal{F}}{\rightrightarrows}} \mathcal{D}$$

the **natural transformation** $\alpha : \mathcal{F} \to \mathcal{G}$, denoted



is a collection of arrows $\{\alpha_X : \mathcal{F}(X) \to \mathcal{G}(X)\}_{X \in \mathrm{Ob}(\mathcal{C})}$ in $\mathcal{D}$ which satisfies **naturality**, i.e. makes the diagram

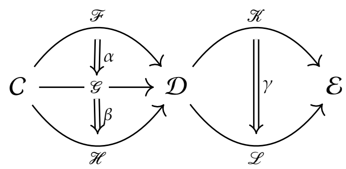


commute for every arrow $f : X \to X'$ in $\mathcal{C}$. The arrows $\{\alpha_X\}_{X \in \mathrm{Ob}(\mathcal{C})}$ are called the **components** of the natural transformation.

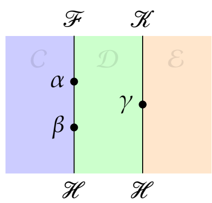
Definition 1.7.2 (Pasting diagram [Nakahira(2023), table 1.1])
A **pasting diagram** represents categories as points (0-dimensional), arrows $\to$

(1-dimensional), natural transformations as surfaces with level-2 arrows ⇒ (0-dimensional).

For example:



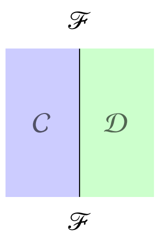
It's dual to a corresponding **string diagram**



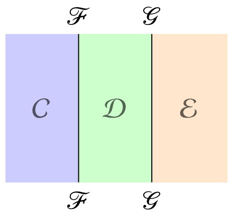
Definition 1.7.3 (Nat [Kostecki(2011), 4.4])
The collection of **natural transformation**s from functors $\mathcal{F}$ to $\mathcal{G}$ is denoted $\mathrm{Nat}(\mathcal{F}, \mathcal{G})$.

Notation 1.7.4 (String diagrams: functor and natural transformation [Marsden(2014), sec. 2])
In **string diagrams**,

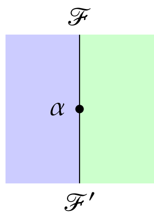
1. A **functor** $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ can be represented as an edge, commonly referred to as a wire:



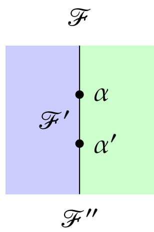
2. Functors compose from left to right:



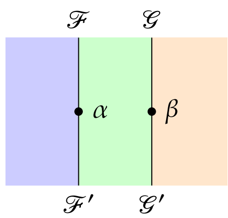
3. A **natural transformation** $\alpha : \mathscr{F} \to \mathscr{F}'$ can be represented as a dot on the wire from top to bottem (the opposite direction of [Marsden(2014)], but the same as [Sterling(2023)]), connecting the two functors :



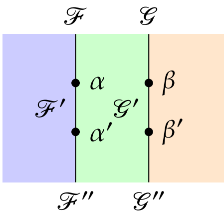
4. Natural transformations (for the same pair of categories) compose vertically from top to bottem:



5. Natural transformations (from different pairs of categories) compose horizontally from left to right:



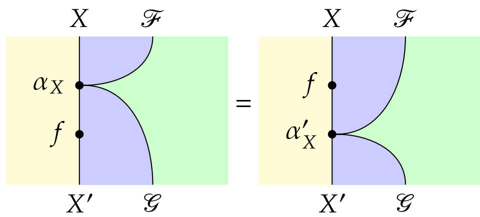
6. The two ways of composing natural transformations are related by the **interchange law**:



i.e.

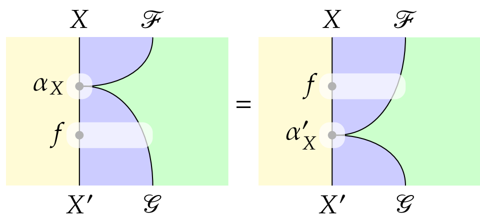
$$(\alpha \bullet \alpha') \bullet (\beta \bullet \beta') = (\alpha \bullet \beta) \bullet (\alpha' \bullet \beta') \tag{1.7.5}$$

7. The **naturality** in natural transformations is equivalent to the following equality:



where X and X' are objects in C, understood as functors from the terminal category $\mathit{1}$ to C.

Since a string diagrams is composed from top to bottem, left to right, we can read



as

$$(X \bullet \mathscr{F}) \bullet (\alpha_X) \bullet (f \bullet \mathscr{G}) = (X' \bullet \mathscr{G}) \quad = \quad (X \bullet \mathscr{F}) \bullet (f \bullet \mathscr{F}) \bullet (\alpha'_X) = (X' \bullet \mathscr{G}) \tag{1.7.6}$$

where each pair of parentheses corresponds to an overlay in the string diagram,

or with the notation in the opposite direction that is more familiar to most:

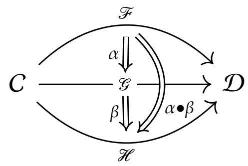
$$\mathscr{G}(f) \circ \alpha_X \circ \mathscr{F}(X) = \mathscr{G}(X') \quad = \quad \alpha'_X \circ \mathscr{F}(f) \circ \mathscr{F}(X) = \mathscr{G}(X') \tag{1.7.7}$$

Note that we read the wire from $\mathscr{F}$ to $\mathscr{G}$ as $\mathscr{F}$ before the natural transformation, but as $\mathscr{G}$ after the transformation.

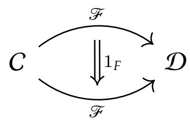
Effectively naturality says that the natural transformation and function f "slide past each other", and so we can draw them as two parallel wires to illustrate this.

Definition 1.7.8 (Functor category [Leinster(2016), 1.3.6])

The **functor category** from $\mathcal{C}$ to $\mathcal{D}$, denoted $[\mathcal{C}, \mathcal{D}]$ or $\mathcal{D}^{\mathcal{C}}$, is a category whose objects are functors from $\mathcal{C}$ to $\mathcal{D}$ and whose arrows are **natural transformation**s between them, where composition is given by



and the identity is given by



Remark 1.7.9 (Indexed, labelled [Kostecki(2011), 4.5])

One can think of a functor category $[\mathcal{C}, \mathcal{D}]$ or $\mathcal{D}^{\mathcal{C}}$ as a category of diagrams in D **indexed** (or **labelled**) by the objects from $\mathcal{C}$.

This particularly makes sense in a **diagram**.

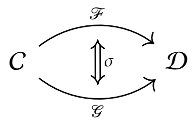
Definition 1.7.10 (Natural isomorphism [Kostecki(2011), 4.2])

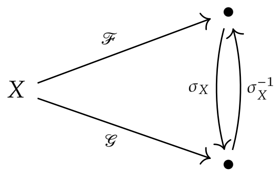
A natural transformation $\sigma : \mathscr{F} \to \mathscr{G}$ between functors $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ and $\mathscr{G} : \mathcal{C} \to \mathcal{D}$ is called a **natural isomorphism** or a **natural equivalence**, denoted $\sigma : \mathscr{F} \cong \mathscr{G}$, if each **component** $\sigma_X : \mathscr{F}(X) \to \mathscr{G}(X)$ is an isomorphism in $\mathcal{D}$, i.e. $\mathscr{F}(X) \underset{\sigma_X}{\cong} \mathscr{G}(X)$.

We call $\mathscr{F}$ and $\mathscr{G}$ **naturally isomorphic** to each other.

We also say that $\mathscr{F}(X) \cong \mathscr{G}(X)$ **naturally in** X [Leinster(2016), 1.3.12].

Diagramatically,





> **Lemma 1.7.11** (Natural isomorphism [Leinster(2016), 1.3.10])
> A **natural isomorphism** between functors from categories $\mathcal{C}$ and $\mathcal{D}$ is an isomorphism in the functor category $[\mathcal{C}, \mathcal{D}]$.

Definition 1.7.12 (Isomorphism of categories [Kostecki(2011), 4.3])

The cateories $\mathcal{C}$ and $\mathcal{D}$ are called **isomorphic**, denoted $\mathcal{C} \cong \mathcal{D}$, iff there exists functors

$$\mathcal{C} \underset{\mathscr{G}}{\overset{\mathscr{F}}{\rightleftarrows}} \mathcal{D}$$

such that

$$1_{\mathcal{C}} = \mathscr{F} \bullet \mathscr{G} \tag{1.7.13}$$

and

$$1_{\mathcal{D}} = \mathscr{G} \bullet \mathscr{F} \tag{1.7.14}$$

Definition 1.7.15 (Equivalence of categories [Kostecki(2011), 4.3])

The categories $\mathcal{C}$ and $\mathcal{D}$ are called **equivalent**, denoted $\mathcal{C} \simeq \mathcal{D}$, iff there exist functors

$$\mathcal{C} \underset{\mathscr{G}}{\overset{\mathscr{F}}{\rightleftarrows}} \mathcal{D}$$

together with **natural isomorphism**s

$$1_{\mathcal{C}} \cong \mathscr{F} \bullet \mathscr{G} \tag{1.7.16}$$

and

$$\mathscr{G} \bullet \mathscr{F} \cong 1_{\mathcal{D}} \tag{1.7.17}$$

1.8 Representables

Remark 1.8.1 (Representables [Leinster(2016), ch. 4, 4.1.15])

A category is a world of objects, all looking at one another. Each sees the world from a different viewpoint.

We may ask: what objects *see*? Fix an object, this can be described by the arrows *from* it, this corresponds to the **covariantly representable functor**.

We can also ask the dual question: how objects are *seen*? Fix an object, this can be described by the arrows *into* it, this corresponds to the **contravariantly representable functor**.

Definition 1.8.2 (Set-valued [Kostecki(2011), 4.4])

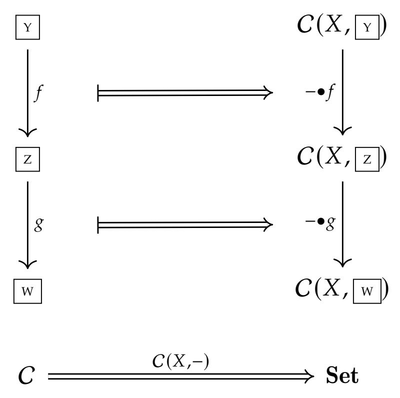
A functor $\mathscr{F} : C \to \mathbf{Set}$ is called **set-valued**.

Definition 1.8.3 (Hom-functor [Kostecki(2011), 3.1, example 10])

For every locally small category C, the **covariant hom-functor**, denoted

$$C(X, -) : C \to \mathbf{Set} \tag{1.8.4}$$

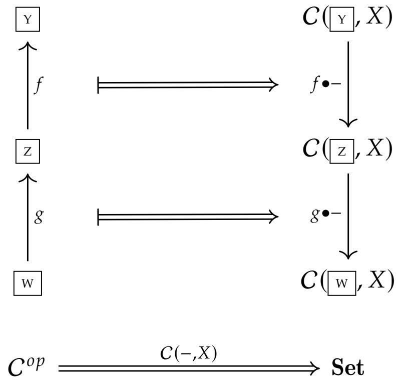
is given by



Conversely, the **contravariant hom-functor**, denoted

$$C(-, X) : C^{op} \to \mathbf{Set} \tag{1.8.5}$$

is given by



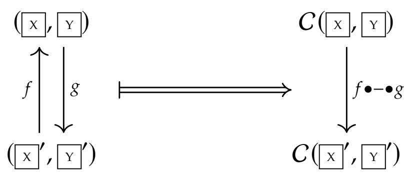
Further, the **hom-bifunctor**, denoted

$$\mathcal{C}(-, =) : \mathcal{C}^{op} \times \mathcal{C} \to \textbf{Set} \tag{1.8.6}$$

defined as a contravariant hom-functor at first argument and as a covariant hom-functor at second argument.

We see − and = as placeholders for any object and its ”associated arrow” (whose domain/codomain is the object, respectively) in the corresponding category. And we use boxes to mark the placeholder objects in diagrams.

Diagramatically [Leinster(2016), 4.1.22],



Definition 1.8.7 (Representable functor [Kostecki(2011), 4.4])
A **set-valued** functor $\mathscr{F} : \mathcal{C} \to \textbf{Set}$ is called **covariantly representable** if for some $X \in \mathcal{C}$,

$$\tau : \mathscr{F} \cong \mathcal{C}(X, -) \tag{1.8.8}$$

where $\cong$ denotes a **natural isomorphism**.

Conversely, a set-valued functor $\mathscr{G} : \mathcal{C}^{op} \to \textbf{Set}$ is called **contravariantly representable** if for some $X \in \mathcal{C}$,

$$\tau : \mathscr{G} \cong C(-, X) \tag{1.8.9}$$

Such an object X is called a **representing object** for the functor $\mathscr{F}$ or $\mathscr{G}$, respectively.

The pair (τ, X) is called a **representation** of the functor $\mathscr{F}$ (respectively, $\mathscr{G}$).

Remark 1.8.10 (Representation [Leinster(2016), 4.1.3, 4.1.17])

A **representation** (τ, X) of a **representable functor** $\mathscr{F}$ is a *choice* of an object $X \in C$ and an isomorphism τ between the corresponding type of **hom-functor** and $\mathscr{F}$.

Representable functors are sometimes just called **representables**. Only set-valued functors can be representable.

Definition 1.8.11 (Yoneda embedding functors [Leinster(2016), 4.1.15])

Let C be a locally small category. The **covariant Yoneda embedding functor** of C is the functor

$$\mathscr{H}^{\bullet} : C^{op} \to [C, \mathbf{Set}] \tag{1.8.12}$$

defined on objects X by the **covariant hom-functor** on X.

This functor embeds what every object in C *sees* the "world" of the category C, i.e. arrows *from* each object.

Conversely, the (contravariant) **Yoneda embedding functor** of C is the functor

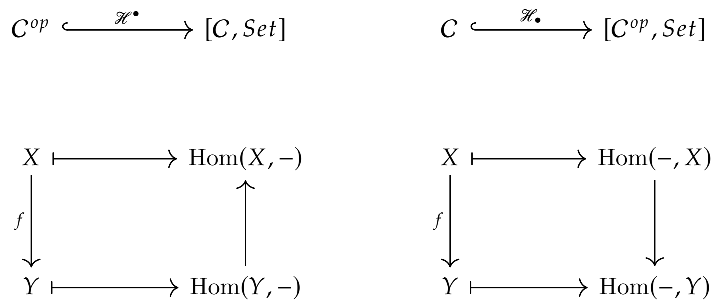
$$\mathscr{H}_{\bullet} : C \to [C^{op}, \mathbf{Set}] \tag{1.8.13}$$

defined on objects X by the **contravariant hom-functor** on X.

This functor embeds how every object in C is "seen", i.e. arrows *to* each object.

$\bullet$ is a placeholder for an object. $\mathscr{H}^X$ and $\mathscr{H}_X$ denote the corresponding Yoneda embedding functors applied to X, and are called covariant/contravariant **Yoneda functor**s, respectively.

Diagramatically [Rosiak(2022), def. 161]:



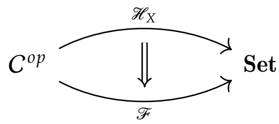
When one speaks of the Yoneda (embedding) functor without specifying covariant or contravariant, it means the contravariant one, because it's the one used in the **Yoneda lemma**.

Lemma 1.8.14 (Yoneda [Leinster(2016), 4.2.1])
Let C be a **locally small** category. Then

$$\mathrm{Nat}(\mathscr{H}_X, \mathscr{F}) \cong \mathscr{F}(X) \tag{1.8.15}$$

naturally in $X \in C$ and $\mathscr{F} \in [C^{op}, \mathbf{Set}]$, where $\mathscr{H}_X$ is the (contravariant) **Yoneda embedding functor** on X, and **Nat** denotes all the natural transformations between the two functors.

Notation 1.8.16 (Yoneda lemma [Leinster(2016), 4.2.1])
Diagramatically, $\mathrm{Nat}(\mathscr{H}_X, \mathscr{F})$ is



and it's also denoted by $[C^{op}, \mathbf{Set}](\mathscr{H}_X, \mathscr{F})$ in the sense of $\mathrm{Hom}_{[C^{op},\mathbf{Set}]}(\mathscr{H}_X, \mathscr{F})$ where $[C^{op}, \mathbf{Set}]$ is a **functor category**.

Remark 1.8.17 (Yoneda philosophy [Rosiak(2022), sec. 6.6])
The Yoneda lemma can be regarded as saying:

> To understand an object it suffices to understand all its relationships with other things.

This is similar to the seventeenth-century philosopher Spinoza's idea that what a body *is* (its "essence") is inseparable from all the ways that the body can affect (causally influence) and be affected (causally influenced) by *other bodies*.

The idea of Yoneda is that we can be assured that if a robot wants to learn whether some object X is the same thing as object Y, it will suffice for it learn whether

$$C(-, X) \cong C(-, Y) \tag{1.8.18}$$

or, dually,

$$C(X, -) \cong C(Y, -) \tag{1.8.19}$$

i.e. whether all the ways of probing X with objects of its environment amount to the same as all the ways of probing Y with objects of its environment.

Lemma 1.8.20 (Full and faithful [Kostecki(2011), 4.8])
The **Yoneda embedding functor** $\mathscr{H}_\bullet : C \to [C^{op}, \mathbf{Set}]$ is **full and faithful**.

Definition 1.8.21 (Ob, Arr [Leinster(2016), 4.1.6])
Given a small category C, there is a functor $\mathrm{Ob} : \mathbf{Cat} \to \mathbf{Set}$ that sends C to its set of objects where **Cat** is the category of small categories. Thus,

$$\mathscr{H}^1(C) \cong \mathrm{Ob}(C) \tag{1.8.22}$$

where $\mathscr{H}$ is a **Yoneda embedding functor**.

This isomorphism is natural in C; hence $\mathrm{Ob} \cong \mathbf{Cat}(1, -)$

where $\mathbf{Cat}(1, -)$ is a **covariant hom-functor**.

Functor Ob is representable. Similarly, the functor $\mathrm{Arr} : \mathbf{Cat} \to \mathbf{Set}$ sending a small category to its set of arrows is representable.

Definition 1.8.23 (Presheaf [Leinster(2016), 1.2.15])
Let C be a category. A **presheaf** $\mathscr{F}$ on C is a functor $C^{op} \to \mathbf{Set}$.

It is called **representable** if $\mathscr{F} \cong \mathscr{H}_X$ for some X.

Notation 1.8.24 (Presheaf)

We'll use $\mathscr{F}$ to denote a **presheaf** since sheaf in French is **faisceau**.

Remark 1.8.25 (Presheaf and Yoneda lemma [Leinster(2016), 4.2.1])

The Yoneda lemma says that for any $X \in C$ and **presheaf** $\mathscr{F}$ on C, a natural transformation $\mathscr{H}_X \to \mathscr{F}$ is an **element** of $\mathscr{F}(X)$ of shape $\mathscr{H}_X$.

We may ask the question [Chen(2016), 68.6.4]:

> What kind of presheaves are already "built in" to the category C?

The answer by the Yoneda lemma is, the Yoneda embedding $\mathscr{H}_\bullet : C \to [C^{op}, \mathbf{Set}]$ embeds C into its own presheaf category.

In mathematics at large, the word "embedding" is used (sometimes informally) to mean a map $i : X \to Y$ that makes X isomorphic to its image in Y, i.e. $X \cong i(X)$. [Leinster(2016), 1.3.19] tells us that in category theory, a full and faithful functor $\mathscr{I} : X \to Y$ can reasonably be called an embedding, as it makes X equivalent to a full subcategory of Y.

So, C is equivalent to the **full subcategory** of the presheaf category $[C^{op}, \mathbf{Set}]$ whose objects are the **representables**.

Definition 1.8.26 (The category of presheaves [Kostecki(2011), 4.5, example 1])

The functor category of contravariant set-valued functors $[C^{op}, \mathbf{Set}]$, called **the category of presheaves** or **varying sets**, the objects of which are contravariant functors $C^{op} \to$ Set. It may be regarded as a category of diagrams in **Set** **indexed** contravariantly by the objects of C.

By definition, objects of C play the role of **stage**s, marking the "positions" (in passive view) or "movements" (in active view) of the varying set $\mathscr{F} : C^{op} \to \mathbf{Set}$. For every X in C^{op}, the set $\mathscr{F}(X)$ is a set of elements of $\mathscr{F}$ at stage X.

An arrow $f : Y \to X$ between two objects in C^{op} induces a **transition arrow** $\mathscr{F}(f) : \mathscr{F}(X) \to \mathscr{F}(Y)$ between the varying set $\mathscr{F}$ at stage A and the varying set $\mathscr{F}$ at stage B.

1.9 Basic types of diagrams

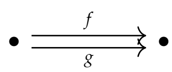
By "basic types of diagrams", we mean some basic structures *inside* a category.

Definition 1.9.1 (Diagram, shape [Leinster(2016), 5.1.18])

Let $\mathcal{C}$ be a category and $\mathcal{J}$ a **small category**. A functor $\mathcal{D} : \mathcal{J} \to \mathcal{C}$ is called a **diagram** in $\mathcal{C}$ of **shape** $\mathcal{J}$.

$\mathcal{J}$ is also called the **indexing category** of the diagram, and we say that $\mathcal{D}$ is a diagram **indexed by** $\mathcal{J}$ [Rosiak(2022), example 35]. J is also called the **template**.

Definition 1.9.2 (Shape E [Leinster(2016), 5.14])
The diagram

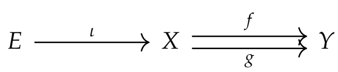


is called a diagram of **shape E**.

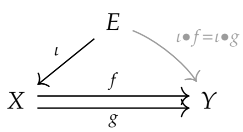
For simplicity, we refer to a diagram of shape E by "a shape $E(f, g)$".

"E" in "shape E" stands for "equal", and the reason will unfold in the definition of **equalizer**.

Definition 1.9.3 (Fork [Leinster(2016), 5.4])
A **fork** over a shape $E(f, g)$ is the diagram



that makes the diagram

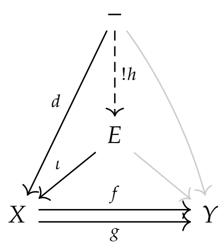


commute.

For simplicity, we refer to a fork by "a fork (E, ι) (over the shape $E(f, g)$)".

Convention 1.9.4 (Grey arrow)
We use **grey arrow**s to represent the composition arrow in a **fork**. This convention is not from literatures and is subject to change.

Definition 1.9.5 (Equalizer [Leinster(2016), 5.1.11])
An **equalizer** of a shape $E(f, g)$ is a fork (E, ι) over it, such that, for any $(-, d)$ over the fork, the diagram



commutes (i.e. any arrow $d : - \longrightarrow X$ must *uniquely* **factor through** E).

For simplicity, we refer to the equalizer of a shape $E(f, g)$ as $\mathrm{Eq}(f, g)$, and ι is the **canonical inclusion**.

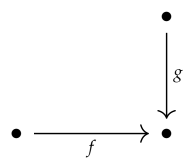
We say that a category $\mathcal{C}$ **has equalizers** iff *every* **shape E** in $\mathcal{C}$ has an equalizer.

Remark 1.9.6 (Equalizing set [Kostecki(2011), eq. 32])
Equalizer in a category is a generalisation of a subset which consists of elements of a given set such that two given functions are equal on them, formally:

For any two arrows $f, g : X \longrightarrow Y$, their **equalizing set** $E \subseteq X$ is defined as

$$E := \{e \mid e \in X \wedge f(e) = g(e)\} \tag{1.9.7}$$

Definition 1.9.8 (Shape P [Leinster(2016), 5.14])
The diagram



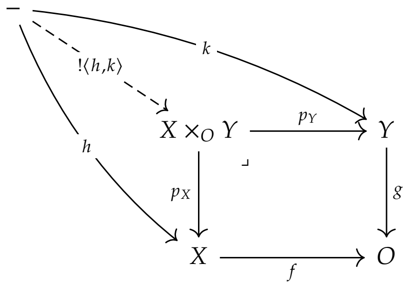
is called a diagram of **shape P**.

For simplicity, we refer to a diagram of shape P by "a shape $P(f, g)$".

"P" in "shape P" may stand for "product/projection/pullback", and the reason will unfold in the definition of **pullback**.

Definition 1.9.9 (Pullback (fiber product) [Kostecki(2011), 2.12])
A **pullback** of a shape $P(f, g)$ is an object $X \times_O Y$ in $\mathcal{C}$ together with arrows p_X

and p_Y, called **projections**, such that, for any object $-$ and arrows h and k, the diagram

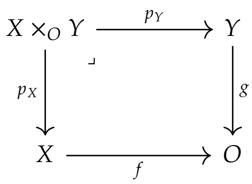


commutes.

We say that a category C **has pullbacks** iff *every* shape $P(f, g)$ in C has a pullback in C.

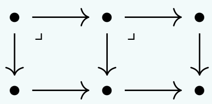
A pullback is also called a **fiber product**.

The square



is called the **pullback square** of f and g. The object $X \times_O Y$ in C is called the **fiber product object**.

Lemma 1.9.10 (Pasting pullbacks [Spivak(2013), 2.5.1.17])
Pullbacks can be pasted together, i.e. for diagram



given that the right-hand square is a pullback, the left-hand square is a pullback if and only if the outer rectangle is a pullback.

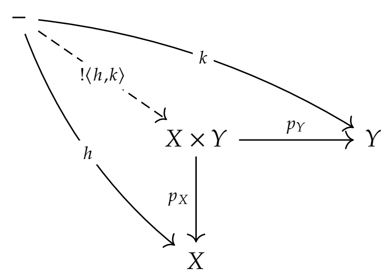
Definition 1.9.11 (Shape T [Leinster(2016), 5.14])
The diagram

is called a diagram of **shape T**.

For simplicity, we refer to a diagram of shape T by "a shape $T(X, Y)$" where X and Y are the 2 objects.

"T" in "shape T" stands for "two". Shape T is useful in the definition of binary product [Kostecki(2011), 2.18].

Definition 1.9.12 (Binary product [Kostecki(2011), 2.18])
A **binary product** of objects X and Y is an object $X \times Y$ in C together with arrows p_X and p_Y, called **projections**, such that, for any object $-$ and arrows h and k, the diagram



commutes.

We say that a category C **has binary products** iff *every* pair X, Y in C has a binary product $X \times Y$ in C.

When there is no confusion, we simply call bianry products **product**s.

Definition 1.9.13 (Coshape, coequalizer, pushout (fiber coproduct), binary coproduct [Kostecki(2011), 2.14, 2.16, 2.19])
coshape, **coequalizer**, **pushout** (**fiber coproduct**), **binary coproduct** can be defined by *reversing all arrows* in the definitions of **shape**, **equalizer**, **pullback (fiber product)**, **binary product** respectively.

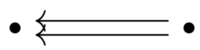
The **pushout** equivalent of the **fiber product object** in pullback is the **fiber coproduct object**, denoted $X +_O Y$, and the **pushout** equivalent of **projection**s in pullback are **injections**, denoted i_X and i_Y, respectively. The unique arrow of a pushout is denoted $[f, g]$.

The binary coproduct equivalent of the **binary product object** in binary product is the **binary coproduct object**, denoted $X + Y$, and the binary coproduct equivalent of **projections** in binary product are **injections**, denoted i_X and i_Y, respectively. The unique arrow of a binary coproduct is denoted $[f, g]$.

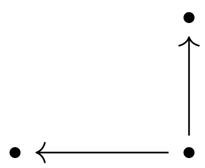
Diagramatically,

- Coshapes:
  - $T =$

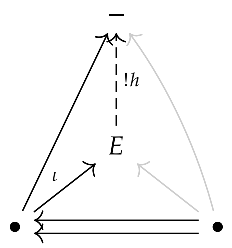
  - $E =$



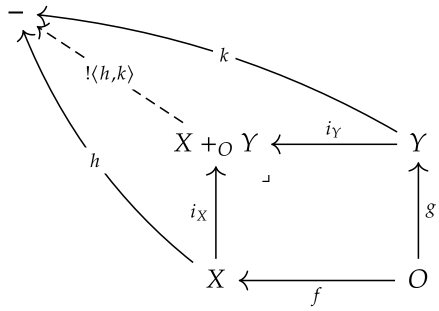
  - $P =$



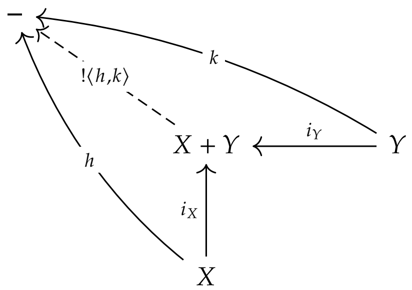
- coequalizer:



- pushout (fiber coproduct):

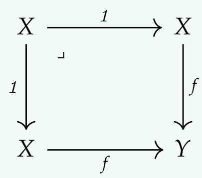


- binary coproduct:



Lemma 1.9.14 (Monic and pullback [Leinster(2016), 5.1.32])

An arrow $X \xrightarrow{f} Y$ is **monic** iff the square



is a **pullback**.

The significance of this lemma is that whenever we prove a result about limits, a result about monics will follow.

Lemma 1.9.15 (Epic and pushout [Leinster(2016), sec. 5.2])

An arrow $X \xrightarrow{f} Y$ is **epic** iff the square

$$\begin{array}{ccc} X & \xleftarrow{1} & X \\ \uparrow{\scriptstyle 1} & \lrcorner & \uparrow{\scriptstyle f} \\ X & \xleftarrow[f]{} & Y \end{array}$$

is a **pushout**.

This is dual to Lemma 1.9.14.

Definition 1.9.16 (N-fold (co)products [Kostecki(2011), 2.22])

In any category with **binary products** the objects $X \times (Y \times Z)$ and $(X \times Y) \times Z$ are isomorphic. In any category with **binary coproducts** the objects $X + (Y + Z)$ and $(X + Y) + Z$ are isomorphic.

This allows to consider ***n*-fold products** $X_1 \times \cdots \times X_n$ and ***n*-fold coproducts** $X_1 + \cdots + X_n$ of objects of a given category.

Definition 1.9.17 (Have finite (co)products [Kostecki(2011), 2.23])

A category which has **n-fold (co)products** for any $n \in \mathbb{N}$ is said to **have finite (co)products**.

1.10 Limits

Remark 1.10.1 (Limits [Leinster(2016), ch. 5])

Adjointness is about the relationships *between* categories. Representability is a property of *set-valued* functors. **Limit**s are about what goes on *inside* a category.

Whenever you meet a method for taking some objects and arrows in a category and constructing a new object out of them, there is a good chance that you are looking at either a **limit** or a **colimit**.

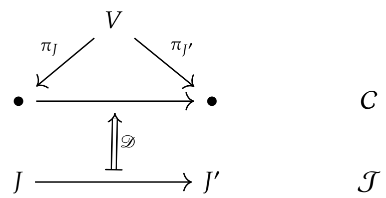
Definition 1.10.2 (Cone [Leinster(2016), 5.1.19])

Let $\mathcal{C}$ be a category, $\mathcal{J}$ a small category, and $\mathcal{D} : \mathcal{J} \to \mathcal{C}$ a **diagram** in $\mathcal{C}$ of shape $\mathcal{J}$.

A **cone** on $\mathcal{D}$ is an object $V \in \mathcal{C}$ (the **vertex** of the cone) together with a family

$$\left(V \xrightarrow{\pi_J} \mathcal{D}(J)\right)_{J \in \mathcal{J}} \tag{1.10.3}$$

of arrows in $\mathcal{C}$ such that for all arrows $J \to J'$ in $\mathcal{J}$, the diagram



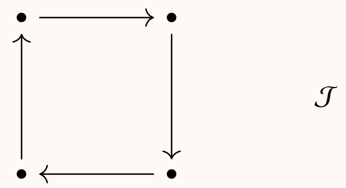
commutes.

The family of arrows are components of a **natural transformation** $\pi : \Delta_V \to \mathcal{D}$, i.e. from the **constant functor** (which assigns the same object V to any object J_i in $\mathcal{J}$) to diagram functor $\mathcal{D}$.

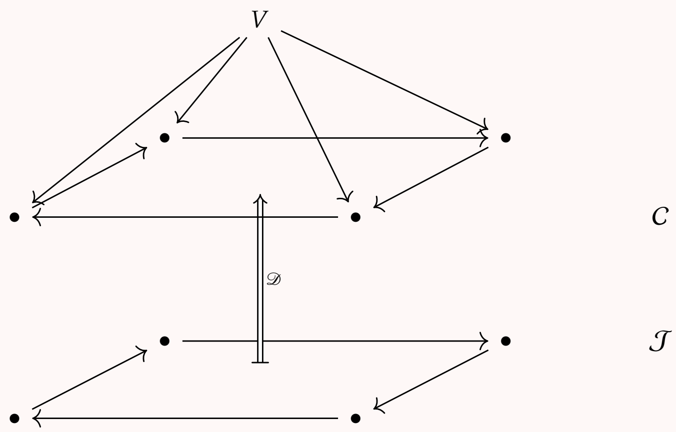
For simplicity, we refer to a cone by "a cone (V, π) on $\mathcal{D}$".

Example 1.10.4 (A cone on a diagram [Kostecki(2011), 4.9])

The cone for a diagram



is

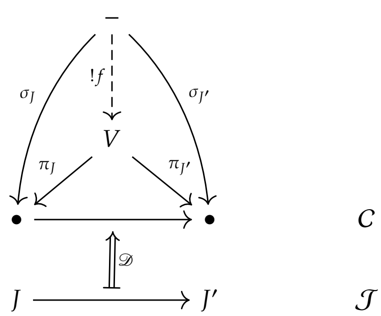


which indeed looks like a cone with the vertex V.

Definition 1.10.5 (Limit [Kostecki(2011), 4.10])
A cone (V, π) on $\mathscr{D} : \mathcal{C} \to \mathcal{J}$ is called a **limit** of $\mathscr{D}$, denoted

$$\lim \mathscr{D} \tag{1.10.6}$$

if the diagram



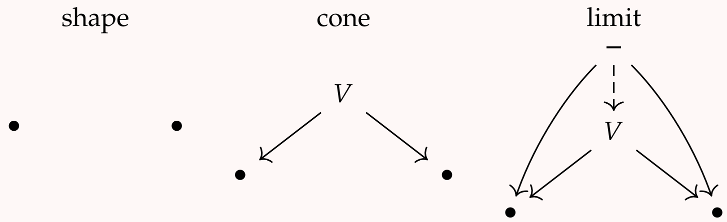
commutes for every cone (V, π) on $\mathscr{D}$.

The arrows π_J are called the **projections** of the **limit**.

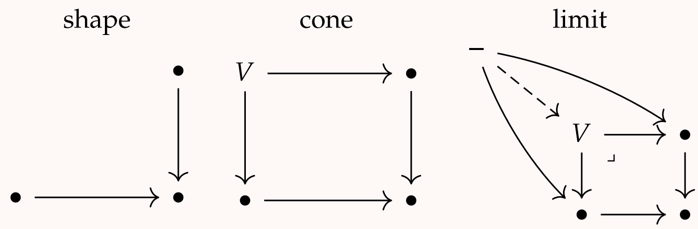
Other possible terms of limit are **limiting cone**, **universal cone**.

Example 1.10.7 (Limits [Kostecki(2011), 4.11, example 1-4])
The basic types of diagrams are actually examples of **limits**:

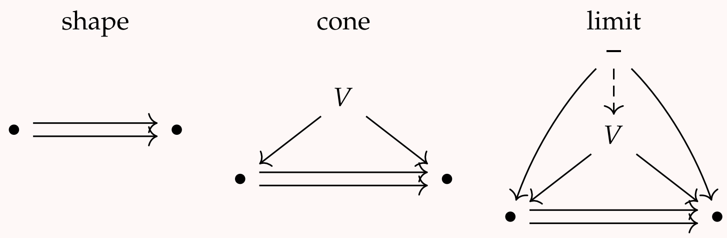
1. binary product [Kostecki(2011), 2.18]:



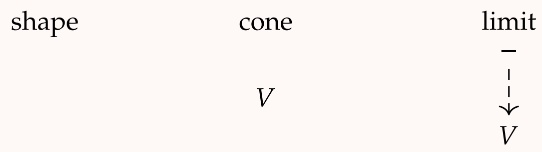
2. pullback (fiber product) [Kostecki(2011), 2.12]:



3. equalizer [Leinster(2016), 5.1.11]:



4. the limit of $\mathscr{D} : \emptyset \to \mathcal{C}$, where $\emptyset$ is an **empty category**:



i.e. the terminal object 1 in $\mathcal{C}$. In particular, for $\mathcal{C} = \mathbf{Set}$ we have

$$- \xrightarrow{!} V = \lim \mathscr{D} = \{\} \tag{1.10.8}$$

Remark 1.10.9 (Cocone, colimit [Kostecki(2011), 4.10])

A **cocone** and a **colimit** are defined by dualization, that is, by reversing the arrows in cone [Leinster(2016), 5.1.19] and limit [Kostecki(2011), 4.10].

In another word, given $\mathscr{D}^{op} : \mathcal{J}^{op} \to \mathcal{C}^{op}$, a cocone on $\mathscr{D}$ is a cone on $\mathscr{D}^{op}$, a colimit of $\mathscr{D}$ is a limit of $\mathscr{D}^{op}$ [Leinster(2016), 5.2.1].

The arrows π_J are called the **coprojection**s of the colimit.

In the same say, one can and show that coequaliser, coproduct, pushout and initial object are examples of colimits.

Lemma 1.10.10 (Limits via products and equalizers [Stacks Project Authors(2017), 002N, 002P])

If all products and equalizers exist, all limits exist.

Dually, if all coproducts and coequalizers exist, all colimits exist.

Definition 1.10.11 (Has (finite) limits, (finitely) complete, left exact [Kostecki(2011), 4.10])

We say that a category $\mathcal{C}$ **has (finite) limits** or is **(finitely) complete** if every diagram $\mathscr{D} : \mathcal{J} \to \mathcal{C}$, where $\mathcal{J}$ is a (finite) category, has a **limit**.

A category $\mathcal{C}$ is called **left exact** iff it is **finitely complete**.

Remark 1.10.12 ((finitely) cocomplete, right exact [Kostecki(2011), 4.10])

Dually to Definition 1.10.11, when every diagram $\mathscr{D} : \mathcal{J} \to C$, where $\mathcal{J}$ is a (finite) category, has a colimit, it is said that the category C **has (finite) colimits** or is **(finitely) cocomplete**.

A category is called **right exact** iff it is **finitely cocomplete**.

Lemma 1.10.13 ((finitely) (co)complete category [Kostecki(2011), 4.14])

A category C is (finitely) complete if it has a terminal object, equalizers and (finite) products, or if it has a terminal object and (finite) pullbacks.

Dually, a category C is (finitely) cocomplete if it has an initial object, co-equalizers and (finite) coproducts, or if it has an initial object and (finite) pushouts.

Lemma 1.10.14 ((co)complete functor category [Kostecki(2011), 4.15])

If category $\mathcal{D}$ is complete and category C is small, then the functor category $\mathcal{D}^C$ is complete.

Dually, if category $\mathcal{D}$ is cocomplete and category C is small, then the functor category $\mathcal{D}^C$ is cocomplete.

Definition 1.10.15 (Preorder, partial order, total order [Kostecki(2011), 1.2, example 9])

Let P be a set. The properties

- (reflexivity) $\forall p \in P, p \leq p$
- (transitivity) $\forall p, q, r \in P, p \leq q \wedge q \leq r \Rightarrow p \leq r$

define a **preorder** $(P, \leq)$.

A **partially ordered** set (called a **partial order**, or a **poset**) is defined as a **preorder** $(P, \leq)$ for which

- (antisymmetry) $\forall p \in P, p \leq q \wedge q \leq p \Rightarrow p = q$

holds.

A **total order** (or a **linear order**) is a **partial order** $(P, \leq)$ for which

- (comparability) $\forall p, q \in P, p \leq q \vee q \leq p$

The category **Preord** consists of objects which are preorders and of arrows which are orderpreserving functions.

The category **Poset** consists of objects which are posets and of arrows which are order-preserving functions between posets, that is, the maps $T : P \longrightarrow P'$ such that

$$p \leq q \Rightarrow T(p) \leq T(q) \tag{1.10.16}$$

Any any **preorder** $(P, \leq)$ and **poset** $(P, \leq)$ can be considered as a category consisting of objects which are elements of a set P and arrows defined by $p \rightarrow q \Longleftrightarrow p \leq q$.

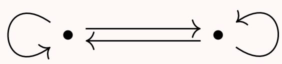
Definition 1.10.17 (Directed poset [Rosiak(2022), def. 285])

A **directed poset** is a **poset** that is inhabited (nonempty) and for which every finite subset has an **upper bound**. Explicitly,

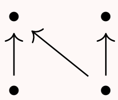
- (directedness) $\forall x, y \in P, \exists z \in P, x \leq z \wedge y \leq z$

Example 1.10.18 (Preorder, poset, directed poset)

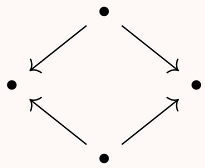
An example of a **preorder** category which is *not* **poset** is:



An example of a **poset** category which is *not* a **directed poset** is [Rosiak(2022), example 3] :

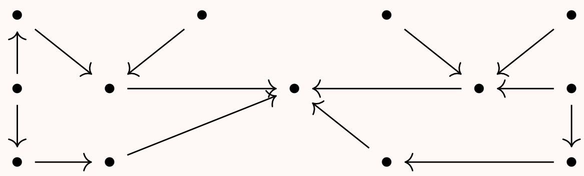


An example that is a **directed poset** category but *not* a **total order** is:



where each pair of nodes has a common upper bound (thus satisfying *directedness*), but there is no path between the two nodes on the center, thus violating *comparability*.

A more complicated example of a **directed poset** category which is *not* a **total order** is [Spivak(2013), example 3.4.1.3]:



One can see immediately that this is a **preorder** because length=0 paths give *reflexivity* and concatenation of paths gives *transitivity*. To see that it is a **partial order** we only note that there are *no loops*.

To see that it is a **poset**, we note that every pair of nodes from one side or both sides has the central node as an upper bound, thus satisfying *directedness*.

But this partial order is *not* a **total order** because there is no path (in either direction) between some nodes, thus violating *comparability*.

Definition 1.10.19 (Inverse limit, projective limit [Kostecki(2011), 4.11])

Let $\mathcal{J}$ be a **directed poset** and $\mathscr{F} : \mathcal{J} \to C$ be a **contravariant functor**. The **limit** of $\mathscr{F}$ is called an **inverse limit** or **projective limit**, and is denoted $\varprojlim_{\mathcal{J}} \mathscr{F}$ or simply $\varprojlim \mathscr{F}$.

Definition 1.10.20 (Direct limit, inductive limit [Kostecki(2011), 4.12])

Let $\mathcal{J}$ be a **directed poset** and $\mathscr{F} : \mathcal{J} \to C$ be a **contravariant functor**. The **colimit** of $\mathscr{F}$ is called a **direct limit** (some called **directed limit**) or **inductive limit**, and is denoted $\varinjlim_{\mathcal{J}} \mathscr{F}$, or simply $\varinjlim \mathscr{F}$.

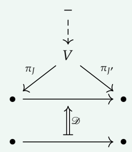
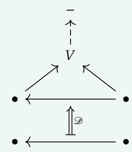
This is dual to **inverse limit**.

Definition 1.10.21 (Preserves (all) (co)limits, left/right exact [Kostecki(2011), 4.13])

A functor $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ **preserves (all) limits** and is called **left exact** iff it sends all limits in $\mathcal{C}$ into limits in $\mathcal{D}$.

Dually, a functor $\mathscr{F} : \mathcal{C} \to \mathcal{D}$ **preserves (all) colimits** and is called **right exact** iff it sends all colimits in $\mathcal{C}$ into colimits in $\mathcal{D}$.

Remark 1.10.22 (Directions in (co)limits)

	Limit	Colimit
diagram		
arrows through the vertex	into the diagram	out of the diagram
on (co)shape P	pullback	pushout
categories have *finite* ...	left exact	right exact
functors preserve all ...	left exact	right exact
on directed poset	inverse/projective limit $\varprojlim \mathscr{F}$	direct/inductive limit $\varinjlim \mathscr{F}$

One can see from the table that, in general, limits have the direction "back" "into" (where "back", "left", "inverse" are directional consistent), and colimits have the opposite: "forward" "out of".

This might help to memorize the directions in these concepts without disorientation.

1.11 Adjunctions

Definition 1.11.1 (Adjoint functor [Kostecki(2011), 5.1])

Given functors

$$\mathcal{C} \underset{\mathscr{R}}{\overset{\mathscr{L}}{\rightleftarrows}} \mathcal{D}$$

we say $\mathscr{L}$ and $\mathscr{R}$ are a pair of **adjoint functors**, or together called an **adjunction** between them, $\mathscr{L}$ is called **left adjoint** to $\mathscr{R}$, and $\mathscr{R}$ is called **right adjoint** to $\mathscr{L}$, denoted

$$\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D} \tag{1.11.2}$$

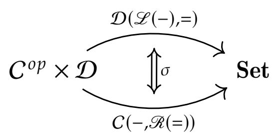
or

$$\mathcal{C} \underset{\mathscr{R}}{\overset{\mathscr{L}}{\underset{\perp}{\rightleftarrows}}} \mathcal{D}$$

iff there exists a **natural isomorphism** σ between the following two **hom-bifunctor**s:

$$\mathcal{D}(\mathscr{L}(-), =) \cong \mathcal{C}(-, \mathscr{R}(=)) \tag{1.11.3}$$

diagramatically,

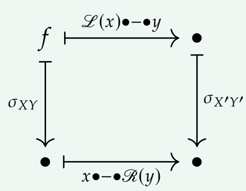


The components of the natural isomorphism σ are isomorphisms

$$\sigma_{XY} : \mathcal{D}(\mathscr{L}(X), Y) \cong \mathcal{C}(X, \mathscr{R}(Y)) \tag{1.11.4}$$

Remark 1.11.5 (Adjoint functor [Kostecki(2011), 5.1])
An adjunction $\mathscr{L} \dashv \mathscr{R}$ means arrows $\mathscr{L}(X) \to Y$ are essentially the same thing as arrows $X \to \mathscr{R}(Y)$ for any $X \in \mathcal{C}$ and $Y \in \mathcal{D}$.

This means the diagram



commutes for any arrows $f : \mathscr{L}(X) \to Y$ in $\mathcal{D}$.

The above can also be diagramatically denoted by **transposition diagram**

$$\frac{X' \quad \xrightarrow{x} \quad X \quad \xrightarrow{\sigma_{XY}(f)} \quad \mathscr{R}(Y) \quad \xrightarrow{\mathscr{R}(y)} \quad \mathscr{R}(Y')}{\mathscr{L}(X') \quad \xrightarrow{\mathscr{L}(x)} \quad \mathscr{L}(X) \quad \xrightarrow{f} \quad Y \quad \xrightarrow{y} \quad Y'} \tag{1.11.6}$$

or simply,

$$\frac{X \to \mathscr{R}(Y) \quad (\mathcal{C})}{\mathscr{L}(X) \to Y \quad (\mathcal{D})} \tag{1.11.7}$$

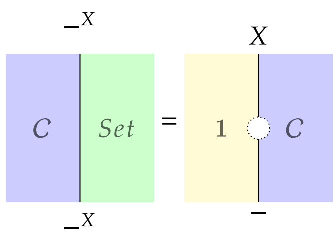
An adjunction is a concept that describes the relationship between two functors that are **weakly inverse** to each other [Nakahira(2023), sec. 4].

By "weakly inverse", we don't mean that applying one after the other gives the identity functor, but in a sense similar to eroding (i.e. enhancing holes) and dilating (i.e. filling holes) an image, applying them in different order yeilds upper/lower "bounds" of the original image [Rosiak(2022), sec. 7.1].

Notation 1.11.8 (String diagrams: adjunction [Nakahira(2023), sec. 3.1])

Here we follow the **string diagram** style of [Marsden(2014)] and [Sterling(2023)], but with additional string diagram types inspired by [Nakahira(2023), eq. 3.1, 4.3].

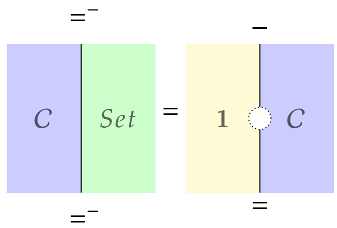
1. The **covariant hom-functor** $\mathcal{C}(X, -)$, denoted $-^X$, can be represented in string diagrams as



   where the dotted circle denotes any arrows with the domain X and codomain $-$.

2. The **contravariant hom-functor** $\mathcal{C}(-, X)$, denoted X^-, can be represented in a similar manner.

3. The **hom-bifunctor** $\mathcal{C}(-, =)$, also denoted $=^-$, can be represented as

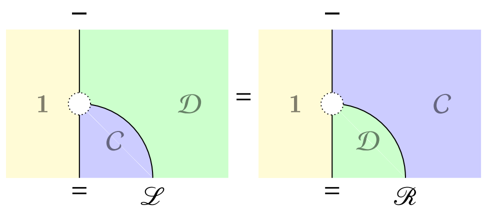


where the dotted circle denotes any arrows with the domain $-$ and codomain $=$.

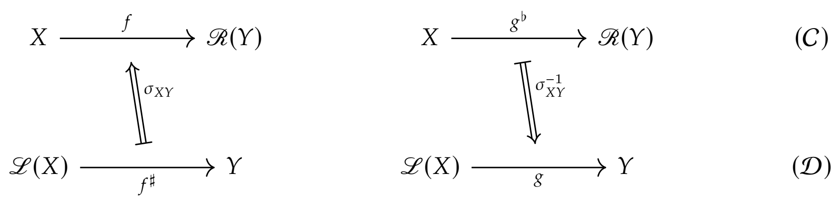
4. The **natural isomorphism**

$$\mathcal{D}(\mathcal{L}(-),=) \cong C(-,\mathcal{R}(=)) \tag{1.11.9}$$

in **adjunction** can be represented as



Definition 1.11.10 (Transpose [Kostecki(2011), 5.1])
Given an adjunction $\mathcal{L} \dashv \mathcal{R} : C \rightleftarrows \mathcal{D}$, there exists $f^\sharp$ and $g^\flat$ such that the diagrams



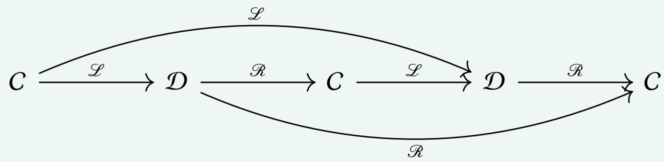
commute for any arrow $f : X \to \mathcal{R}(Y)$ in C, $g : \mathcal{L}(X) \to Y$ in $\mathcal{D}$.

$f^\sharp$ is called the **left transpose** of f. $g^\flat$ is called the **right transpose** of g.

Other possible terms are left/right **adjunct** of each other, and **mates** [nLab(2023)].

Remark 1.11.11 (Idempotent [Zhang(2021), 5.30])

Given an adjunction $\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D}$, we may obtain two **endofunctors** $\mathscr{L} \bullet \mathscr{R} : \mathcal{C} \to \mathcal{C}$ and $\mathscr{R} \bullet \mathscr{L} : \mathcal{D} \to \mathcal{D}$ that commute the diagram



that means they are both **idempotent**, i.e. applying $\mathscr{L} \bullet \mathscr{R}$ any times yields the same result as applying it once, and similarly for $\mathscr{R} \bullet \mathscr{L}$.

Definition 1.11.12 ((co)unit [Zhang(2021), 5.30])

Given an adjunction $\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D}$, the natural transformation

$$\eta : 1_{\mathcal{C}} \to \mathscr{L} \bullet \mathscr{R} \tag{1.11.13}$$

is called the **unit** of the adjunction, and

$$\epsilon : \mathscr{R} \bullet \mathscr{L} \to 1_{\mathcal{D}} \tag{1.11.14}$$

is called the **counit**.

We call an arrow

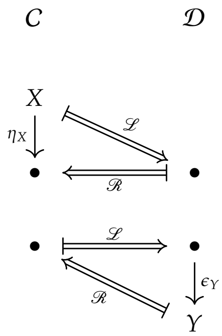
$$\eta_X : X \to (\mathscr{L} \bullet \mathscr{R})(X) \tag{1.11.15}$$

a **unit** over X, and

$$\epsilon_Y : (\mathscr{R} \bullet \mathscr{L})(Y) \to Y \tag{1.11.16}$$

a **counit** over Y. They are components of the natural transformations η and ϵ, respectively.

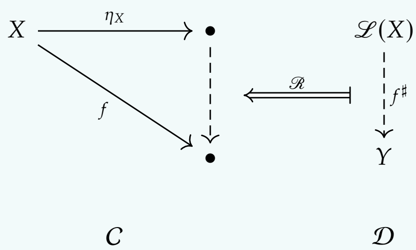
Diagramatically, the diagrams



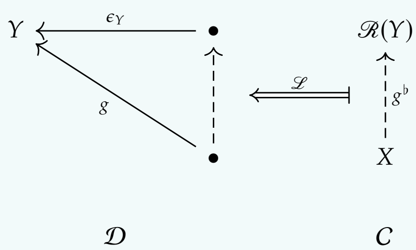
commute.

Lemma 1.11.17 (Universality of (co)unit [Kostecki(2011), 5.3])

The unit η and counit ϵ of an adjunction $\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D}$ are universal, i.e. the diagram



commutes for any $f \in \mathcal{C}$, and the diagram



commutes for any $g \in \mathcal{D}$.

Lemma 1.11.18 (Triangle identities [Kostecki(2011), 5.4])

Given an adjunction $\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D}$, the diagrams

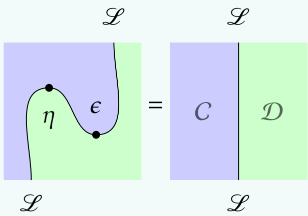
$$
\begin{array}{ccc}
\mathscr{L} & \xrightarrow{\mathscr{L}(\eta)} & \mathscr{L} \bullet \mathscr{R} \bullet \mathscr{L} \\
 & {\scriptstyle 1_{\mathscr{L}}} \searrow & \downarrow {\scriptstyle \epsilon_{\mathscr{L}}} \\
 & & \mathscr{L}
\end{array}
\qquad
\begin{array}{ccc}
\mathscr{R} & \xrightarrow{\eta_{\mathscr{R}}} & \mathscr{R} \bullet \mathscr{L} \bullet \mathscr{R} \\
 & {\scriptstyle 1_{\mathscr{R}}} \searrow & \downarrow {\scriptstyle \mathscr{R}(\epsilon)} \\
 & & \mathscr{R}
\end{array}
$$

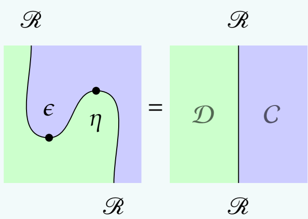
commute.

Note that $\mathscr{L}$ in $\epsilon_{\mathscr{L}}$ is a subscript, meaning $\epsilon_{\mathscr{L}} : \mathcal{D} \to \mathcal{D}, \mathscr{L}(X) \mapsto \epsilon_{\mathscr{L}(X)}$ for $X \in \mathcal{C}$. Similar for $\eta_{\mathscr{R}}$.

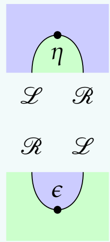
Lemma 1.11.19 (Snake identities [Nakahira(2023), thm. 4.8])

Continuing from Notation 1.11.8, the **triangle identities** can be represented in **string diagrams** as follows, and called the **snake identities** (or **zig-zag identities**):



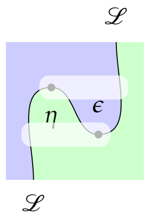


where



are the unit and counit of the adjunction, respectively.

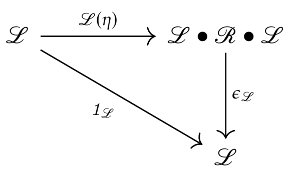
Notation 1.11.20 (String diagram: snake identities [Nakahira(2023), thm. 4.8])
Following Notation 1.7.4, recall that a string diagram is composed from top to bottom, left to right, we can read the left snake



in **snake identities** as

$$\mathscr{L} \xmapsto{(\eta \bullet \mathscr{L}) \bullet (\mathscr{L} \bullet \epsilon)} \mathscr{L} \tag{1.11.21}$$

where each pair of parentheses corresponds to an overlay in the string diagram, and this is equivalent to



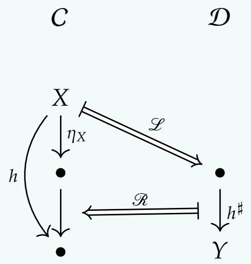
in **triangle identities**.

Lemma 1.11.22 ((co)unit and transposes [Leinster(2016), 2.2.4])

Given an adjunction

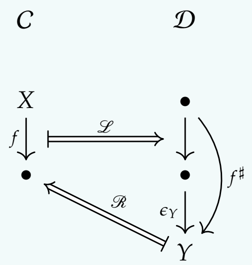
$$\mathcal{C} \underset{\mathscr{R}}{\overset{\mathscr{L}}{\rightleftarrows}} \mathcal{D}$$

with unit η and counit ϵ, the diagrams



$$h = \eta_X \bullet \mathscr{R}(h^\sharp) \tag{1.11.23}$$

and

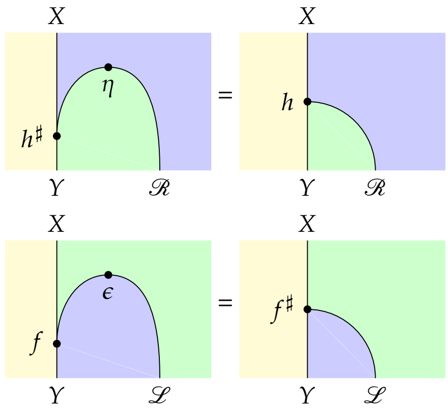


$$f^\sharp = \mathscr{L}(f) \bullet \epsilon_Y \tag{1.11.24}$$

commute.

Notation 1.11.25 (String diagrams: (co)unit and transposes [Marsden(2014), lem. 3.6])

In **string diagrams**, Lemma 1.11.22 can be represented as:



Remark 1.11.26 (Topologically plausible [Leinster(2016), 2.2.9])

The **string diagrams** in Lemma 1.11.19 and Notation 1.11.25 are **topologically plausible** equations, i.e. the equality can be obtained by simply pulling the string straight.

Lemma 1.11.27 ((co)unit and natural isomorphism [Kostecki(2011), eq. 127])

The natural transformation σ_{XY} and τ_{XY} that are the components of the natural isomorphism in the adjunction $\mathscr{L} \dashv \mathscr{R} : \mathcal{C} \rightleftarrows \mathcal{D}$ are related to the unit and counit of the adjunction:

$$\begin{aligned} \sigma_{XY}(f) &= \eta_X \bullet \mathscr{R}(f^\sharp) \\ \tau_{XY}(g) &= \mathscr{L}(g^\flat) \bullet \epsilon_Y \end{aligned} \tag{1.11.28}$$

and they are reverse of each other

$$\sigma_{XY} = \tau_{YX}^{-1} \tag{1.11.29}$$

Proof. This can be read out from the diagrams in universality of (co)unit [Kostecki(2011), 5.3] and transpose [Kostecki(2011), 5.1]. □

Lemma 1.11.30 (Uniqueness of adjoints [Kostecki(2011), 5.8])
A left or right adjoint, if it exists, is unique up to natural isomorphism.

Proof. For the left adjoint, from the university of η it follows that there exists a unique, up to isomorphism, isomorphism between different left adjoints. It remains to show naturality of this isomorphim, which is left as an exercise. The proof for right adjoint follows by duality. □

Theorem 1.11.31 (Adjunction via (co)units [Leinster(2016), 2.2.5])
Given categories and functors

$$\mathcal{C} \underset{\mathscr{R}}{\overset{\mathscr{L}}{\rightleftarrows}} \mathcal{D}$$

there is a *one-to-one correspondence* between the **adjunction** $\mathscr{L} \dashv \mathscr{R}$ and the pairs of **natural transformations** η and ϵ satisfying the the **triangle identities**.

Proof. From Lemma 1.11.18, it follows that every adjunction between $\mathscr{L}$ and $\mathscr{R}$ gives rise to a pair of transformations η and ϵ satisfying the triangle identities.

To show that there exists a unique adjunction for η and ϵ, the uniqueness follows from Lemma 1.11.22, the existence can use the construction in the spirit of Definition 1.11.10. □

Theorem 1.11.32 (Adjunction via initial objects [Leinster(2016), 2.3.6])
Given categories and functors

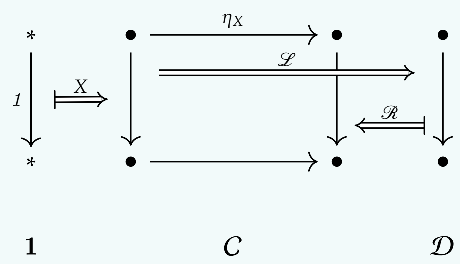
$$\mathcal{C} \underset{\mathscr{R}}{\overset{\mathscr{L}}{\rightleftarrows}} \mathcal{D}$$

there is a *one-to-one correspondence* between:

1. the **adjunction** $\mathscr{L} \dashv \mathscr{R}$

2. natural transformations $\eta : 1_C \to \mathscr{L} \bullet \mathscr{R}$ such that η_X is **initial** in the **comma category** $X \Rightarrow \mathscr{R}$ for every $X \in C$

   Diagramatically,



   where the functor X is the **constant object functor**.

1.12 Interactions

Remark 1.12.1 (Interactions [Leinster(2016)])

In this section, we will discuss the interactions between

- (co)limits
- adjunctions
- representables

and their relationships with universal properties.

Lemma 1.12.2 (Adjunction preserves (co)limits [Leinster(2016), 6.3.1])

Given an adjunction $\mathscr{L} \dashv \mathscr{R} : C \rightleftarrows \mathcal{D}$, $\mathscr{L}$ preserves colimits, and $\mathscr{R}$ preserves limits.

Explictly, given $\mathscr{D} : \mathcal{J} \to C$, we have

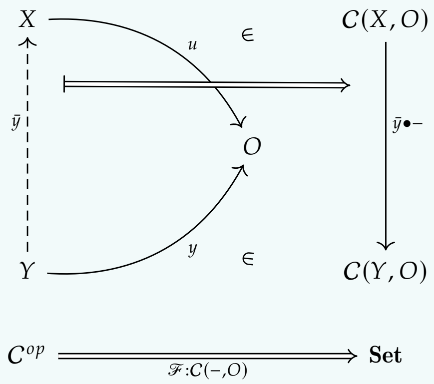
$$(\operatorname{colim} \mathscr{D}) \bullet \mathscr{L} \cong \operatorname{colim}(\mathscr{D} \bullet \mathscr{L}) \tag{1.12.3}$$

and given $\mathscr{D}' : \mathcal{J} \to \mathcal{D}$, we have

$$(\lim \mathscr{D}') \bullet \mathscr{R} \cong \lim(\mathscr{D}' \bullet \mathscr{R}) \tag{1.12.4}$$

Corollary 1.12.5 (A representation is a universal element [Leinster(2016), 4.3.2])

Let C be a locally small category and $\mathscr{F} : C^{op} \longrightarrow \mathbf{Set}$. Then a **representation** of $\mathscr{F}$ consists of a pair (X, u) such that the diagram



commutes.

Remark 1.12.6 (Universal element [Leinster(2016), 4.3.2])

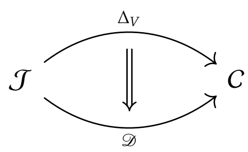
Pairs (Y, y) with $Y \in C$ and $y \in \mathscr{F}(Y)$ in Corollary 1.12.5 are sometimes called **element**s of the **presheaf** $\mathscr{F}$.

Indeed, Lemma 1.8.14 (Yoneda) tells us that y amounts to a **generalized element** of $\mathscr{F}$ of shape $\mathscr{H}_Y$.

An element u satisfying condition in Corollary 1.12.5 is sometimes called a **universal element** of $\mathscr{F}$. So, Corollary 1.12.5 says that a representation of a presheaf $\mathscr{F}$ amounts to a universal element of $\mathscr{F}$.

Lemma 1.12.7 (Adjunction and representable [Leinster(2016), 4.1.11])
Any set-valued functor with a left adjoint is representable.

Definition 1.12.8 (Cone as a natural transformation [Leinster(2016), eq. 6.1])
Now, given a diagram $\mathscr{D} : \mathcal{J} \to \mathcal{C}$ and an object $V \in \mathcal{C}$, a **cone** on $\mathscr{D}$ with vertex V is simply a natural transformation from the **diagonal functor** Δ_V to the diagram $\mathscr{D}$.



Writing $\mathrm{Cone}(V, \mathscr{D})$ for the set of cones on $\mathscr{D}$ with vertex V, we therefore have

$$\mathrm{Cone}(V, \mathscr{D}) = [\mathcal{J}, \mathcal{C}](\Delta_V, \mathscr{D}). \tag{1.12.9}$$

Thus, $\mathrm{Cone}(V, \mathscr{D})$ is **functorial in** V (contravariantly) and $\mathscr{D}$ (covariantly).

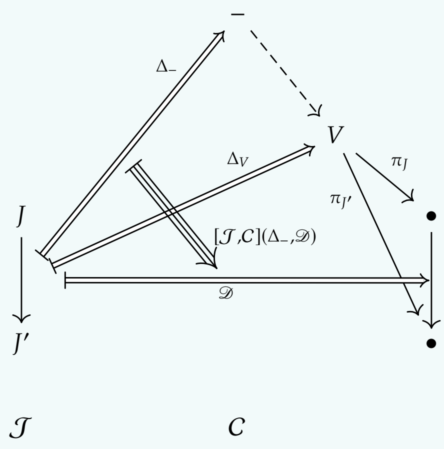
Lemma 1.12.10 (Limit via representation [Leinster(2016), 6.1.1])
Let $\mathcal{J}$ be a small category, $\mathcal{C}$ a category, and $\mathscr{D} : \mathcal{J} \to \mathcal{C}$ a diagram. Then there is a one-to-one correspondence between

- limit cones on $\mathscr{D}$
- representations of the natural transformation **Cone**

with the representing objects being the limit objects (i.e. the vertices) of $\mathscr{D}$.

Briefly put: a **limit** (V, π) of $\mathscr{D}$ is a **representation** of $[\mathcal{J}, \mathcal{C}](\Delta_{-}, \mathscr{D})$.

Diagramatically,



It implies that

$$\mathrm{Cone}(-, \mathscr{D}) \cong \mathcal{C}\left(-, \lim_{\leftarrow \mathcal{J}} \mathscr{D}\right) \tag{1.12.11}$$

for any $- \in \mathcal{C}$.

Lemma 1.12.12 (Representables preserve limits [Leinster(2016), 6.2.2])
Let $\mathscr{A}$ be a locally small category and $X \in \mathcal{C}$. Then $\mathcal{C}(X, -) : \mathcal{C} \to \mathbf{Set}$ preserves limits.

Proof. It follows from Lemma 1.12.10 and that [Leinster(2016), 6.2.1]

$$\mathrm{Cone}(X, \mathscr{D}) \cong \lim_{\leftarrow \mathcal{J}} \mathcal{C}(X, \mathscr{D}) \tag{1.12.13}$$

naturally in X and $\mathscr{D}$. □

Lemma 1.12.14 (Limits commute with limits [Leinster(2016), 6.2.8])

Let $\mathcal{I}$ and $\mathcal{J}$ be small categories. Let $\mathcal{C}$ be a locally small category with limits of shape $\mathcal{I}$ and of shape $\mathcal{J}$.

Define

$$\begin{array}{rccc} \mathscr{D}^\bullet : & \mathcal{I} & \to & [\mathcal{J}, \mathcal{C}] \\ & I & \mapsto & \mathscr{D}(I, -) \end{array} \tag{1.12.15}$$

and

$$\begin{array}{rccc} \mathscr{D}_\bullet : & \mathcal{J} & \to & [\mathcal{I}, \mathcal{C}] \\ & J & \mapsto & \mathscr{D}(-, J) \end{array} \tag{1.12.16}$$

Then for all $\mathscr{D} : \mathcal{I} \times \mathcal{J} \to \mathcal{C}$, we have

$$\lim_{\leftarrow \mathcal{J}} \lim_{\leftarrow \mathcal{I}} \mathscr{D}^\bullet \cong \lim_{\leftarrow \mathcal{I}\mathcal{J}} \mathscr{D} \cong \lim_{\leftarrow \mathcal{I} \leftarrow \mathcal{J}} \mathscr{D}_\bullet \tag{1.12.17}$$

and all these limits exist. In particular, $\mathcal{C}$ has limits of shape $\mathcal{I} \times \mathcal{J}$.

Lemma 1.12.18 (Colimits commute with colimits [Leinster(2016), 6.2.10])

Dual to Lemma 1.12.14, colimits commute with colimits.

Remark 1.12.19 ([Leinster(2016), 6.2.10])

Limits do *not* in general commute with colimits.

Some special cases where they do:

- **filtered colimit**s commute with finite limits [Stacks Project Authors(2017), 002W].

Lemma 1.12.20 (Initial and terminal objects via adjunction [Leinster(2016), 2.1.9])

Initial and terminal objects can be described as adjoints. Let $\mathcal{C}$ be a category. There exist the **unique functor** $! : \mathcal{C} \to \mathbf{1}$, and a **constant object functor** $X : 1 \to \mathcal{C}$ for each object X.

A **left adjoint** to ! is exactly an **initial object** of $\mathcal{C}$:

$$0 \dashv ! : \mathbf{1} \rightleftarrows \mathcal{C} \tag{1.12.21}$$

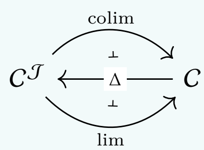
Similarly, a **right adjoint** to ! is exactly a **terminal object** of $\mathcal{C}$:

$$! \dashv 1 : \mathcal{C} \rightleftarrows \mathbf{1} \tag{1.12.22}$$

Proof. In both cases, being an adjunction gives an isomorphism for each object X, one side of the isomorphism is $\mathbf{1}(*, *)$ which is just 1_*, and the other side are $\mathcal{C}(0, X)$ or $\mathcal{C}(X, 1)$, and the isomorphism establishes the uniqueness of the arrows (from 0 or to 1) for each object. The initial or terminal object exists if the corresponding adjunction exists. □

Lemma 1.12.23 ((co)limits via adjunction [Rosiak(2022), example 200])

(Co)limits can be phrased entirely in terms of adjunctions:



The advantages of this adjunction perspective is that the (co)limit of *every* $\mathcal{J}$-shaped diagram in $\mathcal{C}$ can be defined all at once.

1.13 Cartesian closed categories

Definition 1.13.1 (Evaluation [Leinster(2016), 6.2.4])

Let $\mathcal{S}$ be a small category, $\mathcal{C}$ a locally small category. For each $X \in \mathcal{S}$, there is a functor

$$\begin{array}{rccc} \mathrm{ev}_X : & [\mathcal{S},\mathcal{C}] & \to & \mathcal{C} \\ & \mathscr{F} & \mapsto & \mathscr{F}(X) \end{array} \tag{1.13.2}$$

called **evaluation** at X.

Given a diagram $\mathscr{D} : \mathcal{J} \to [\mathcal{S},\mathcal{C}]$, we have for each $X \in \mathcal{S}$ a functor

$$\begin{array}{rccc} \mathscr{D} \bullet \mathrm{ev}_X : & \mathcal{J} & \to & \mathcal{C} \\ & J & \mapsto & \mathscr{D}(J)(X) \end{array} \tag{1.13.3}$$

We write $\mathscr{D} \bullet \mathrm{ev}_X$ as $\mathscr{D}(-)(X)$.

Definition 1.13.4 (Cartesian product functor [Kostecki(2011), 5.13])
If $\mathcal{C}$ is a category with **binary products**, we can define for every X the **cartesian product functor** $X \times (-) : \mathcal{C} \to \mathcal{C}$, with the following action:

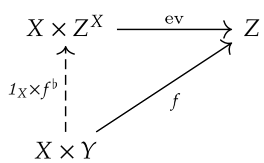
$$\begin{aligned} (X \times (-))(Y) &= X \times Y \\ (X \times (-))(f) &= \mathrm{id}_X \times f \end{aligned} \tag{1.13.5}$$

Definition 1.13.6 (Exponential [Kostecki(2011), 5.13])
If for $X \in \mathcal{C}$, a **cartesian product functor** $X \times (-)$ has a right adjoint, it is called an **exponential** or the **exponential object**, denoted $(-)^X$.

Explicitly, $X \times (-) \dashv (-)^X$ means there is a **natural isomorphism** of bifunctors $\mathrm{Hom}(X \times (-), -) \cong \mathrm{Hom}(-, (-)^X)$, i.e. for any arrow $f : X \times Y \to Z$ there is a unique arrow $f^\flat : Y \to Z^X$, which is the **transpose** of the adjunction, called **exponential transpose**.

The arrow $\mathrm{ev} : X \times (-)^X \to (-)$ is called the **evaluation** arrow of the exponential, and is the **counit** of the adjunction.

Diagramatically, the diagram



commutes.

We say that category $\mathcal{C}$ **has exponentials** if for any $X \in \mathcal{C}$, there exists an **exponential** $(-)^X$.

Definition 1.13.7 (Cartesian closed category [Kostecki(2011), 5.14])

A category C is called **cartesian closed** iff C **has exponentials** and **has finite products**.

Definition 1.13.8 (Power object [Kostecki(2011), 6.8])

The **power object** $P(X)$ of an object X in a **cartesian closed category** C with **subobject classifier** Ω is defined as the **exponential object** Ω^X.

If Ω^X exists for any X in C, we say that C **has power objects**.

Lemma 1.13.9 (Properties of cartesian closed categories [Kostecki(2011), 5.15])

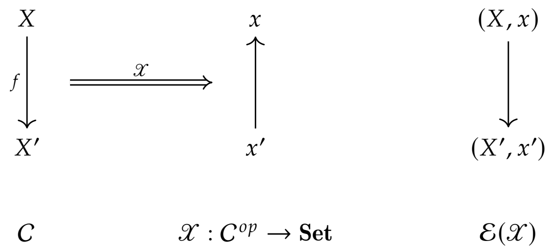
For any **cartesian closed category** C, and any objects X, Y, Z of C, we have

1. $0 \times X \cong 0$ if 0 exists in C
2. $1 \times X \cong X$
3. $X^0 \cong 1$ if 0 exists in C
4. $X^1 \cong X$
5. $1^X \cong 1$
6. $X \times Y \cong Y \times X$
7. $(X \times Y) \times Z \cong X \times (Y \times Z)$
8. $Y^X \times Z^X \cong (Y \times Z)^X$
9. $Z^{X \times Y} \cong \left(Z^X\right)^Y$
10. $X + Y \cong Y + X$
11. $(X + Y) + Z \cong X + (Y + Z)$
12. $Z^X \times Z^Y \cong Z^{X+Y}$ if C has binary coproducts
13. $(X \times Y) + (X \times Z) \cong X \times (Y + Z)$ if C has binary coproducts

1.14 Subobject classifier

Definition 1.14.1 (Category of elements [Leinster(2016), 6.2.16])

Let C be a category and $\mathcal{X} : C^{op} \to \mathbf{Set}$ a **presheaf** on C. The **category of elements** $\mathcal{E}(\mathcal{X})$ of $\mathcal{X}$ is the category in which:



There is a **projection functor** $\mathscr{P} : \mathbf{E}(\mathscr{X}) \to C$ defined by $\mathscr{P}(X, x) = X$ and $\mathscr{P}(f) = f$.

The Yoneda lemma implies that for a presheaf $\mathscr{X}$, the **generalized element**s of $\mathscr{X}$ of representable shape correspond to objects of the category of elements.

Theorem 1.14.2 (Density [Leinster(2016), 6.2.17])

Let C be a small category and $\mathscr{X}$ a presheaf on C. Then $\mathscr{X}$ is the colimit of the diagram

$$\mathcal{E}(\mathscr{X}) \xrightarrow{\mathscr{P}} C \xrightarrow{H_\bullet} [C^{op}, \text{Set}] \tag{1.14.3}$$

in $[C^{op}, \mathbf{Set}]$, i.e.

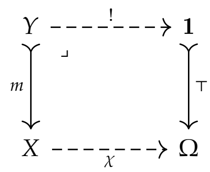
$$\mathscr{X} \cong \lim_{\to \mathcal{E}(\mathscr{X})} (\mathscr{P} \bullet H_\bullet) \tag{1.14.4}$$

where $\mathcal{E}(\mathscr{X})$ is the **category of elements**, $\mathscr{P}$ the **projection functor** in it, and $H_\bullet$ the (contravariant) **Yoneda embedding**.

This theorem states that every presheaf is a colimit of representables in a canonical way. It is secretly dual to the Yoneda lemma. This becomes apparent if one expresses both in suitably lofty categorical language (that of ends, or that of bimodules).

Definition 1.14.5 (Subobject classifier [Kostecki(2011), 6.4])

A **subobject classifier** is an object Ω in C, together with an arrow $\top : \mathbf{1} \to \Omega$, called the **true arrow**, such that for each monic arrow $m : Y \rightarrowtail X$ there is a *unique* arrow $\chi : X \to \Omega$, called the **characteristic arrow** of m (or of Y), such that the diagram



is a **pullback**, where ! is the **unique functor**.

Ω is also called a **generalized truth-value object**.

The arrow $(! \bullet T) : Y \xrightarrow{!} \mathbf{1} \rightarrowtail \Omega$ is often denoted as $\top_Y : Y \to \Omega$.

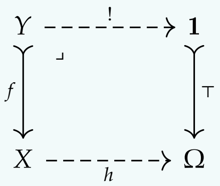
Lemma 1.14.6 (Isomorphism to class of subobjects [Kostecki(2011), 6.7])
In any category $\mathcal{C}$ with a **subobject classifier** Ω,

$$\mathrm{Sub}(X) \cong \mathcal{C}(X, \Omega) \tag{1.14.7}$$

i.e. the **class of subobjects** of an object X in $\mathcal{C}$ is isomorphic to the class of arrows from X to Ω.

Proof. It follows from the definitions and Lemma 1.9.14 that for every $f : Y \to X$ and $[f] \in \mathrm{Sub}(X)$,

- (surjection) $\chi(f) \in \mathcal{C}(X, \Omega)$
- (injection) for every $h : X \to \Omega$, $\chi(f) = h$ since



is a pullback. □

Definition 1.14.8 (Power object functor [Kostecki(2011), 6.8])
The contravariant **power object functor** $\mathscr{P} : \mathcal{C}^{op} \to \mathcal{C}$, given by

$$\mathscr{P} : X \mapsto \Omega^X \tag{1.14.9}$$

for $X \in \mathrm{Ob}(\mathcal{C})$

and such that $\mathscr{P}(f) : \Omega^Y \mapsto \Omega^X$ for $f : X \to Y$ in $\mathcal{C}$ is given by

$$\mathscr{P}(f)(Y) = \{x \in X \mid f(x) \in Y\} \tag{1.14.10}$$

When power object is defined for cartesian closed categories, we have

$$\frac{X \times Y \to \Omega}{Y \to \Omega^X} \tag{1.14.11}$$

thus for every category with power objects

$$\mathrm{Hom}(X \times Y, \Omega) \cong \mathrm{Hom}\left(Y, \Omega^X\right) \tag{1.14.12}$$

This equation, together with Lemma 1.14.6 written in the form $\mathrm{Sub}(X \times Y) \cong \mathrm{Hom}(X \times Y, \Omega)$, gives the isomorphism

$$\mathrm{Sub}(X \times Y) \cong \mathrm{Hom}(Y, \mathscr{P}(X)) \tag{1.14.13}$$

2 Topos theory

Definition 2.0.1 (Topos [Kostecki(2011), 7.1])
A **topos** or **elementary topos** is a category satisfying one of these equivalent conditions:

- it is a **complete** category with **exponential**s and **subobject classifier**
- it is a **complete** category with **subobject classifier** and its **power object**
- it is a **cartesian closed** category with **equalizer**s and **subobject classifier**

Since the **completeness** of a category with subobject classifier implies its **co-completeness**. Thus a topos not only **has all finite limits**, but also **has all finite colimits**.

This means that topos is such category which has, in particular,

1. terminal object
2. equalizers
3. pullbacks
4. all other limits
5. exponential objects
6. subobject classifier

Remark 2.0.2 (Topoi, toposes [Kostecki(2011), 7.1])

The name "topos" originates from the Greek word "τοπος", meaning a *place*, as topos could mean a place of geometry, and at the same time as a place of logic.

Following the ancient Greek naming convention, the plural of topos is **topoi**, but people also use **toposes**. We use them interchangeably.

Example 2.0.3 (Topos)

Topos theory unifies, in an extraordinary way, important aspects of geometry and logic.

Grothendieck topos was first introduced by Grothendieck to generalize **topological space** [Kostecki(2011), 7.2]:

> every space gives rise to a topos (namely, the category of sheaves on it).

Topological properties of the space can be reinterpreted in a useful way as categorical properties of its associated topos.

Elementary topos was introduced by Lawvere and Tierney, to generalize **Set**. A topos can be regarded as a 'universe of sets' [Leinster(2016), 6.3.20]. **Set** is the most basic example of a topos, and every topos shares enough features with Set that [source]

> anything you really really *needed* to do in the category of sets can be done in any topos.

Every **presheaf** category, is a topos. [Leinster(2016), 6.3.27]

A topos can allow the interpretation of a higher-order logic. In particular, objects can be seen as collections of elements of a given **type**, subobjects are viewed as **propositions**. Products and coproducts are interpreted as conjunction and disjunction respectively. For an introduction, see [Pitts(2001)].

Definition 2.0.4 (Geometric morphism, Topoi [Kostecki(2011), 7.2])

If $\mathcal{E}_1$ and $\mathcal{E}_2$ are **topos**es, then a **geometric morphism** $\mathscr{G} : \mathcal{E}_1 \to \mathcal{E}_2$ is defined as a pair of **adjoint functors** $\mathscr{G}^* \dashv \mathscr{G}_*$ between $\mathcal{E}_1$ and $\mathcal{E}_2$, such that $\mathscr{G}^*$ **preserves**

finite limits (i.e. is **left exact**), which implies $\mathscr{G}_*$ **preserves colimits** (i.e. is **right exact**).

The category of toposes and their geometric morphisms is denoted **Topoi**.

3 Appendix

These are my notes on:

- Category theory
- Topos theory
- Type theory
- Sheaf theory
- Differential sheaves
- SDG (Synthetic Differential Geometry)

The primary references for these notes are:

- [Kostecki(2011)] for a clean introduction from category theory to topos theory
- [Leinster(2016)] for its breakdown and simplification of category theory
- [Kostecki(2009)] for its introduction to SDG
- [Rosiak(2022)] for its sheaf examples
- [Mallios and Zafiris(2015)] for its introduction to Differential sheaves
- [Rosiak(2022)] for its examples of sheaves
- [Zhang(2021)] for a friendly introduction to type theory using the language of category theory
- [Chen(2016)] for various preliminaries on category theory
- [Sterling(2023)] for its introduction to models of type theory, and extensive use of string diagrams in the style of [Marsden(2014)]
- [Fauser(2004)] for the use of Kuperberg graphical calculi over commutative diagrams

For draft notes, see drafts for Notes on Topos Theory and Type Theory.

References

[Chen(2016)] Evan Chen. 2016. An infinitely large napkin. (Cited in sections 1.8.25 and 3)

[Fauser(2004)] Bertfried Fauser. 2004. Grade free product formulae from Graßmann-Hopf gebras. *Clifford Algebras: Applications to Mathematics, Physics, and Engineering* (2004), 279–301. https://arxiv.org/abs/math-ph/0208018 (Cited in section 3)

[Fong and Spivak(2018)] Brendan Fong and David I Spivak. 2018. Seven sketches in compositionality: An invitation to applied category theory. *arXiv preprint arXiv:1803.05316* (2018). https://arxiv.org/abs/1803.05316 (Cited in sections 1.1.4 and 1.6.2)

[Freyd(1976)] Peter Freyd. 1976. Properties invariant within equivalence types of categories. In *Algebra, Topology, and Category Theory*. Elsevier, 55–61. http://angg.twu.net/Freyd76.html (Cited in section 1.6.2)

[Kostecki(2009)] Ryszard Paweł Kostecki. 2009. Differential Geometry in Toposes. https://www.fuw.edu.pl/~kostecki/sdg.pdf (Cited in section 3)

[Kostecki(2011)] Ryszard Paweł Kostecki. 2011. An Introduction to Topos Theory. *Technial Report* (2011). https://www.fuw.edu.pl/~kostecki/ittt.pdf (Cited in sections 1.1.1, 1.1.4, 1.1.5, 1.1.11, 1.1.12, 1.2.1, 1.2.2, 1.2.3, 1.2.5, 1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.3.5, 1.3.6, 1.3.8, 1.3.13, 1.4.1, 1.4.2, 1.4.5, 1.4.6, 1.4.7, 1.5.1, 1.5.2, 1.5.5, 1.5.8, 1.5.10, 1.6.1, 1.6.4, 1.7.3, 1.7.9, 1.7.10, 1.7.12, 1.7.15, 1.8.2, 1.8.3, 1.8.7, 1.8.20, 1.8.26, 1.9.6, 1.9.9, 1.9.11, 1.9.12, 1.9.13, 1.9.16, 1.9.17, 1.10.4, 1.10.5, 1.10.7, 1, 2, 1.10.9, 1.10.11, 1.10.12, 1.10.13, 1.10.14, 1.10.15, 1.10.19, 1.10.20, 1.10.21, 1.11.1, 1.11.5, 1.11.10, 1.11.17, 1.11.18, 1.11.27, 1.11.27, 1.11.30, 1.13.4, 1.13.6, 1.13.7, 1.13.8, 1.13.9, 1.14.5, 1.14.6, 1.14.8, 2.0.1, 2.0.2, 2.0.3, 2.0.4, and 3)

[Leinster(2016)] Tom Leinster. 2016. Basic category theory. *arXiv preprint arXiv:1612.09375* (2016). https://arxiv.org/abs/1612.09375 (Cited in sections 1.1.7, 1.3.4, 1.3.9, 1.3.10, 1.3.11, 1.3.12, 1.3.16, 1.3.18, 1.4.3, 1.5.3, 1.5.5, 1.5.9, 1.6.2, 1.6.3, 1.7.1, 1.7.8, 1.7.10, 1.7.11, 1.8.1, 1.8.3, 1.8.10, 1.8.11, 1.8.14, 1.8.16, 1.8.21, 1.8.23, 1.8.25, 1.9.1, 1.9.2, 1.9.3, 1.9.5, 1.9.8, 1.9.11, 1.9.14, 1.9.15, 1.10.1, 1.10.2, 3, 1.10.9, 1.11.22, 1.11.26, 1.11.31, 1.11.32, 1.12.1, 1.12.2, 1.12.5, 1.12.6, 1.12.7, 1.12.8, 1.12.10, 1.12.12, 1.12.14, 1.12.18, 1.12.19, 1.12.20, 1.13.1, 1.14.1, 1.14.2, 2.0.3, and 3)

[Mallios and Zafiris(2015)] Anastasios Mallios and Elias Zafiris. 2015. *Differential sheaves and connections: a natural approach to physical geometry*. Vol. 18. World Scientific. (Cited in section 3)

[Marsden(2014)] Daniel Marsden. 2014. Category theory using string diagrams. *arXiv preprint arXiv:1401.7220* (2014). (Cited in sections 1.1.14, 1.7.4, 3, 1.11.8, 1.11.25, and 3)

[Nakahira(2023)] Kenji Nakahira. 2023. Diagrammatic category theory. *arXiv preprint arXiv:2307.08891* (2023). (Cited in sections 1.1.4, 1.1.12, 1.1.13, 1.5.4, 1.5.5, 1.7.2, 1.11.5, 1.11.8, 1.11.19, and 1.11.20) [nLab(2020)] nLab. 2020. universal construction - nLab. https://ncatlab.org/nlab/show/universal+construction (Cited in section 1.6.2)

[nLab(2023)] nLab. 2023. adjunct - nLab. https://ncatlab.org/nlab/show/adjunct (Cited in section 1.11.10)

[Ochs(2022)] Eduardo Ochs. 2022. On the the missing diagrams in Category Theory (first-person version). *arXiv preprint arXiv:2204.10630* (2022). https://arxiv.org/abs/2204.10630 (Cited in section 1.6.2)

[Pitts(2001)] Andrew M Pitts. 2001. Categorical logic. *Handbook of logic in computer science* 5 (2001), 39–128. (Cited in section 2.0.3)

[Riehl(2017)] Emily Riehl. 2017. *Category theory in context*. Courier Dover Publications. (Cited in section 1.6.2)

[Rosiak(2022)] Daniel Rosiak. 2022. *Sheaf theory through examples*. MIT Press. (Cited in sections 1.8.11, 1.8.17, 1.9.1, 1.10.17, 1.10.18, 1.11.5, 1.12.23, and 3)

[Spivak(2013)] David I Spivak. 2013. Category theory for scientists. *arXiv preprint arXiv:1302.6946* (2013). (Cited in sections 1.9.10 and 1.10.18)

[Stacks Project Authors(2017)] The Stacks Project Authors. 2017. Stacks Project. http://stacks.math.columbia.edu (Cited in sections 1.10.10 and 1.12.19)

[Sterling(2023)] Jon Sterling. 2023. Notes on models of type theory. https://www.jonmsterling.com/jms-00DJ.xml (Cited in sections 3, 1.11.8, and 3)

[Zhang(2021)] Tesla Zhang. 2021. Type theories in category theory. *arXiv preprint arXiv:2107.13242* (2021). (Cited in sections 1.1.7, 1.11.11, 1.11.12, and 3)

Alphabetical Index