1

Introduction to Categories and Categorical Logic

Samson Abramsky and Nikos Tzevelekos

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.

Preface

The aim of these notes is to provide a succinct, accessible introduction to some of the basic ideas of category theory and categorical logic. The notes are based on a lecture course given at Oxford over the past few years. They contain numerous exercises, and hopefully will prove useful for self-study by those seeking a first introduction to the subject, with fairly minimal prerequisites. The coverage is by no means comprehensive, but should provide a good basis for further study; a guide to further reading is included.

The main prerequisite is a basic familiarity with the elements of discrete mathematics: sets, relations and functions. An Appendix contains a summary of what we will need, and it may be useful to review this first. In addition, some prior exposure to abstract algebra — vector spaces and linear maps, or groups and group homomorphisms — would be helpful.

Contents

1.1 Introduction

Why study categories — what are they good for? We can offer a range of answers for readers coming from different backgrounds:

- For **mathematicians**: category theory organises your previous mathematical experience in a new and powerful way, revealing new connections and structure, and allows you to "think bigger thoughts".
- For **computer scientists**: category theory gives a precise handle on important notions such as compositionality, abstraction, representation-independence, genericity and more. Otherwise put, it provides the fundamental mathematical structures underpinning many key programming concepts.
- For **logicians**: category theory gives a syntax-independent view of the fundamental structures of logic, and opens up new kinds of models and interpretations.
- For **philosophers**: category theory opens up a fresh approach to structuralist foundations of mathematics and science; and an alternative to the traditional focus on set theory.

- For **physicists**: category theory offers new ways of formulating physical theories in a structural form. There have *inter alia* been some striking recent applications to quantum information and computation.

1.1.1 From Elements To Arrows

Category theory can be seen as a "generalised theory of functions", where the focus is shifted from the pointwise, set-theoretic view of functions, to an abstract view of functions as *arrows*.

Let us briefly recall the arrow notation for functions between sets.[1] A function f with *domain* X and *codomain* Y is denoted by: $f : X \to Y$.

$$\text{Diagrammatic notation: } X \xrightarrow{f} Y \,.$$

The fundamental operation on functions is *composition*: if $f : X \to Y$ and $g : Y \to Z$, then we can define $g \circ f : X \to Z$ by $g \circ f(x) = g(f(x))$. Note that, in order for the composition to be defined, the codomain of f must be the same as the domain of g.

$$\text{Diagrammatic notation: } X \xrightarrow{f} Y \xrightarrow{g} Z \,.$$

Moreover, for each set X there is an *identity function* on X, which is denoted by:

$$\mathsf{id}_X : X \longrightarrow X \qquad \mathsf{id}_X(x) = x \,.$$

These operations are governed by the *associativity law* and the *unit laws*. For $f : X \to Y$, $g : Y \to Z$, $h : Z \to W$:

$$(h \circ g) \circ f = h \circ (g \circ f) \,, \qquad f \circ \mathsf{id}_X = f = \mathsf{id}_Y \circ f \,.$$

Notice that these equations are formulated purely in terms of the algebraic operations on functions, without any reference to the elements of the sets X, Y, Z, W. We will refer to any concept pertaining to functions which can be defined purely in terms of composition and identities as *arrow-theoretic*. We will now take a first step towards learning to "think with arrows" by seeing how we can replace some familiar definitions couched in terms of elements by arrow-theoretic equivalents; this will lead us towards the notion of category.

We say that a function $f : X \longrightarrow Y$ is:

injective if $\forall x, x' \in X.\, f(x) = f(x') \implies x = x'$,
surjective if $\forall y \in Y.\, \exists x \in X.\, f(x) = y$,

monic if $\forall g, h.\, f \circ g = f \circ h \implies g = h$,
epic if $\forall g, h.\, g \circ f = h \circ f \implies g = h$.

[1] A review of basic ideas about sets, functions and relation, and some of the notation we will be using, is provided in Appendix A.

Note that injectivity and surjectivity are formulated in terms of elements, while epic and monic are arrow-theoretic.

Proposition 1. *Let $f : X \to Y$. Then,*

1. *f is injective iff f is monic.*
2. *f is surjective iff f is epic.*

Proof: We show 1. Suppose $f : X \to Y$ is injective, and that $f \circ g = f \circ h$, where $g, h : Z \to X$. Then for all $z \in Z$:

$$f(g(z)) = f \circ g(z) = f \circ h(z) = f(h(z)) .$$

Since f is injective, this implies $g(z) = h(z)$. Hence we have shown that

$$\forall z \in Z.\ g(z) = h(z) ,$$

and so we can conclude that $g = h$. So f injective implies f monic.
For the converse, fix a one-element set $\mathbf{1} = \{\bullet\}$. Note that elements $x \in X$ are in 1–1 correspondence with functions $\bar{x} : \mathbf{1} \to X$, where $\bar{x}(\bullet) = x$. Moreover, if $f(x) = y$ then $\bar{y} = f \circ \bar{x}$. Writing injectivity in these terms, it amounts to the following:

$$\forall x, x' \in X.\ f \circ \bar{x} = f \circ \bar{x}' \implies \bar{x} = \bar{x}'.$$

Thus we see that being injective is a *special case* of being monic. ■

Exercise 1. Show that $f : X \to Y$ is surjective iff it is epic.

1.1.2 Categories Defined

Definition 1 A ***category*** $\mathcal{C}$ consists of:

- A collection $\mathsf{Ob}(\mathcal{C})$ of ***objects***. Objects are denoted by A, B, C, etc.
- A collection $\mathsf{Ar}(\mathcal{C})$ of ***arrows*** (or ***morphisms***). Arrows are denoted by f, g, h, etc.
- Functions $\mathsf{dom}, \mathsf{cod} : \mathsf{Ar}(\mathcal{C}) \longrightarrow \mathsf{Ob}(\mathcal{C})$, which assign to each arrow f its ***domain*** $\mathsf{dom}(f)$ and its ***codomain*** $\mathsf{cod}(f)$. An arrow f with domain A and codomain B is written $f : A \to B$. For each pair of objects A, B, we define the set
$$\mathcal{C}(A, B) := \{f \in \mathsf{Ar}(\mathcal{C}) \mid f : A \to B\} .$$
We refer to $\mathcal{C}(A, B)$ as a ***hom-set***. Note that distinct hom-sets are *disjoint*.
- For any triple of objects A, B, C, a ***composition*** map
$$c_{A,B,C} : \mathcal{C}(A, B) \times \mathcal{C}(B, C) \longrightarrow \mathcal{C}(A, C) .$$
$c_{A,B,C}(f, g)$ is written $g \circ f$ (or sometimes $f; g$). Diagrammatically: $A \xrightarrow{f} B \xrightarrow{g} C$.
- For each object A, an ***identity*** morphism $\mathsf{id}_A : A \to A$.

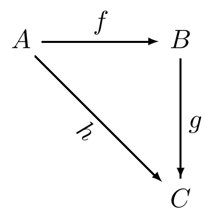
The above must satisfy the following axioms:

$$h \circ (g \circ f) = (h \circ g) \circ f\,, \qquad f \circ \mathsf{id}_A = f = \mathsf{id}_B \circ f\,.$$

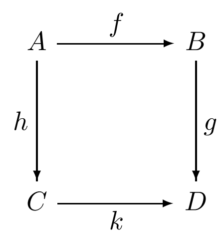
whenever the domains and codomains of the arrows match appropriately so that the compositions are well-defined. ▲

1.1.3 Diagrams in Categories

Diagrammatic reasoning is an important tool in category theory. The basic cases are commuting triangles and squares. To say that the following triangle commutes



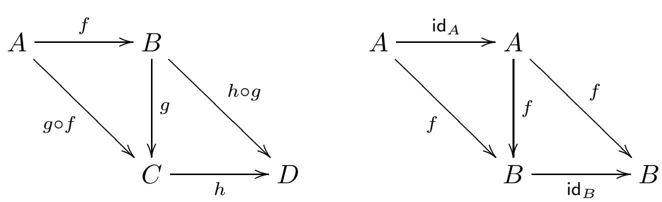
is exactly equivalent to asserting the equation $g \circ f = h$. Similarly, to say that the following square commutes



means exactly that $g \circ f = k \circ h$. For example, the equations

$$h \circ (g \circ f) = (h \circ g) \circ f\,, \qquad f \circ \mathsf{id}_A = f = \mathsf{id}_B \circ f\,,$$

can be expressed by saying that the following diagrams commute.



As these examples illustrate, most of the diagrams we shall use will be "pasted together" from triangles and squares: the commutation of the diagram as a whole will then reduce to the commutation of the constituent triangles and squares.

We turn to the general case. The formal definition is slightly cumbersome; we give it anyway for reference.

Definition 2 We define a ***graph*** to be a collection of ***vertices*** and ***directed edges***, where each edge $e : v \to w$ has a specified source vertex v and target vertex w. Thus graphs are like categories without composition and identities.[2] A ***diagram in a category*** $\mathcal{C}$ is a graph whose vertices are labelled with objects of $\mathcal{C}$ and whose edges are labelled with arrows of $\mathcal{C}$, such that, if $e : v \to w$ is labelled with $f : A \to B$, then we must have v labelled by A and w labelled by B. We say that such a diagram *commutes* if any two paths in it with common source and target, and at least one of which has length greater than 1, are equal. That is, given paths

$$A \xrightarrow{f_1} C_1 \xrightarrow{f_2} \cdots C_{n-1} \xrightarrow{f_n} B \qquad \text{and} \qquad A \xrightarrow{g_1} D_1 \xrightarrow{g_2} \cdots D_{m-1} \xrightarrow{g_m} B,$$

if $\max(n, m) > 1$ then

$$f_n \circ \cdots \circ f_1 = g_m \circ \cdots \circ g_1 \,.$$

▲

To illustrate this definition, to say that the following diagram commutes

$$E \xrightarrow{\quad e \quad} A \underset{g}{\overset{f}{\rightrightarrows}} B$$

amounts to the assertion that $f \circ e = g \circ e$; it does *not* imply that $f = g$.

1.1.4 Examples

Before we proceed to our first examples of categories, we shall present some background material on partial orders, monoids and topologies, which will provide running examples throughout these notes.

Partial orders

A *partial order* is a structure $(P, \leq)$ where P is a set and $\leq$ is a binary relation on P satisfying:

- $x \leq x$ (Reflexivity)
- $x \leq y \ \wedge y \leq x \ \ \Rightarrow \ \ x = y$ (Antisymmetry)
- $x \leq y \ \wedge \ y \leq z \ \ \Rightarrow \ \ x \leq z$ (Transitivity)

For example, $(\mathbb{R}, \leq)$ and $(\mathcal{P}(X), \subseteq)$ are partial orders, and so are strings with the sub-string relation.

If P, Q are partial orders, a map $h : P \longrightarrow Q$ is a *partial order homomorphism* (or *monotone function*) if:

$$\forall x, y \in P.\ x \leq y \implies h(x) \leq h(y)\,.$$

Note that homomorphisms are closed under composition, and that identity maps are homomorphisms.

[2] This would be a "multigraph" in normal parlance, since multiple edges between a given pair of vertices are allowed.

Monoids

A *monoid* is a structure $(M, \cdot, 1)$ where M is a set,

$$\_ \cdot \_ : M \times M \longrightarrow M$$

is a binary operation, and $1 \in M$, satisfying the following axioms:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)\,, \qquad 1 \cdot x = x = \cdot 1\,.$$

For example, $(\mathbb{N}, +, 0)$ is a monoid, and so are strings with string-concatenation. Moreover, groups are special kinds of monoids.

If M, N are monoids, a map $h : M \to N$ is a *monoid homomorphism* if

$$\forall m_1, m_2 \in M.\, h(m_1 \cdot m_2) = h(m_1) \cdot h(m_2)\,, \qquad h(1) = 1\,.$$

Exercise 2. Suppose that G and H are groups (and hence monoids), and that $h : G \longrightarrow H$ is a monoid homomorphism. Prove that h is a group homomorphism.

Topological spaces

A topological space is a pair (X, T_X) where X is a set, and T_X is a family of subsets of X such that

- $\varnothing, X \in T_X$,
- if $U, V \in T_X$ then $U \cap V \in T_X$,
- if $\{U_i\}_{i \in I}$ is any family in T_X, then $\bigcup_{i \in I} U_i \in T_X$.

A *continuous map* $f : (X, T_X) \to (Y, T_Y)$ is a function $f : X \to Y$ such that, for all $U \in T_Y$, $f^{-1}(U) \in T_X$.

Let us now see some first examples of categories.

- Any kind of mathematical structure, together with structure preserving functions, forms a category. E.g.
  - **Set** (sets and functions)
  - **Mon** (monoids and monoid homomorphisms)
  - **Grp** (groups and group homomorphisms)
  - $\mathbf{Vect}_k$ (vector spaces over a field k, and linear maps)
  - **Pos** (partially ordered sets and monotone functions)
  - **Top** (topological spaces and continuous functions)
- **Rel**: objects are sets, arrows $R : X \to Y$ are *relations* $R \subseteq X \times Y$. Relational composition:

$$R; S(x, z) \iff \exists y.\, R(x, y) \,\wedge\, S(y, z)$$

- Let k be a field (for example, the real or complex numbers). Consider the following category $\mathbf{Mat}_k$. The objects are natural numbers. A morphism $M : \mathbf{n} \longrightarrow \mathbf{m}$ is an $\mathbf{n} \times \mathbf{m}$ matrix with entries in k. Composition is matrix multiplication, and the identity on $\mathbf{n}$ is the $\mathbf{n} \times \mathbf{n}$ diagonal matrix.
- ◇ Monoids are one-object categories. Arrows correspond to the elements of the monoid, with the monoid operation being arrow-composition and the monoid unit being the identity arrow.
- ◇ A category in which for each pair of objects A, B there is at most one morphism from A to B is the same thing as a ***preorder***, i.e. a reflexive and transitive relation.

Note that our first class of examples illustrate the idea of categories as *mathematical contexts*; settings in which various mathematical theories can be developed. Thus for example, **Top** is the context for general topology, **Grp** is the context for group theory, etc.

On the other hand, the last two examples illustrate that many important mathematical structures themselves appear as categories of particular kinds. The fact that two such different kinds of structures as monoids and posets should appear as extremal versions of categories is also rather striking.

This ability to capture mathematics both "in the large" and "in the small" is a first indication of the flexibility and power of categories.

Exercise 3. Check that **Mon**, $\mathbf{Vect}_k$, **Pos** and **Top** are indeed categories.

Exercise 4. Check carefully that monoids correspond exactly to one-object categories. Make sure you understand the difference between such a category and **Mon**. (For example: how many objects does **Mon** have?)

Exercise 5. Check carefully that preorders correspond exactly to categories in which each homset has at most one element. Make sure you understand the difference between such a category and **Pos**. (For example: how big can homsets in **Pos** be?)

1.1.5 First Notions

Many important mathematical notions can be expressed at the general level of categories.

Definition 3 Let $\mathcal{C}$ be a category. A morphism $f : X \to Y$ in $\mathcal{C}$ is:

- ***monic*** (or a ***monomorphism***) if $f \circ g = f \circ h \implies g = h\,,$
- ***epic*** (or an ***epimorphism***) if $g \circ f = h \circ f \implies g = h\,.$

An ***isomorphism*** in $\mathcal{C}$ is an arrow $i : A \to B$ such that there exists an arrow $j : B \to A$ — the ***inverse*** of i — satisfying

$$j \circ i = \mathsf{id}_A\,, \qquad i \circ j = \mathsf{id}_B\,.$$

We denote isomorphisms by $i : A \xrightarrow{\cong} B$, and write i^{-1} for the inverse of i. We say that A and B ***are isomorphic***, $A \cong B$, if there exists some $i : A \xrightarrow{\cong} B$▲

Exercise 6. Show that the inverse, if it exists, is unique.

Exercise 7. Show that $\cong$ is an equivalence relation on the objects of a category.

As we saw previously, in **Set** monics are injections and epics are surjections. On the other hand, isomorphisms in **Set** correspond exactly to bijections, in **Grp** to group isomorphisms, in **Top** to homeomorphisms, in **Pos** to order isomorphisms, etc.

Exercise 8. Verify these claims.

Thus we have at one stroke captured the key notion of isomorphism in a form which applies to *all* mathematical contexts. This is a first taste of the level of generality which category theory naturally affords.

We have already identified monoids as one-object categories. We can now identify groups as *exactly those one-object categories in which every arrow is an isomorphism.* This also leads to a natural generalization, of considerable importance in current mathematics: a ***groupoid*** is a category in which every morphism is an isomorphism.

Opposite Categories and Duality

The directionality of arrows within a category $\mathcal{C}$ can be reversed without breaking the conditions of category; this yields the notion of ***opposite category***.

Definition 4 Given a category $\mathcal{C}$, ***the opposite category*** $\mathcal{C}^{\mathsf{op}}$ is given by taking the same objects as $\mathcal{C}$, and

$$\mathcal{C}^{\mathsf{op}}(A, B) := \mathcal{C}(B, A)\,.$$

Composition and identities are inherited from $\mathcal{C}$. ▲

Note that if we have

$$A \xrightarrow{f} B \xrightarrow{g} C$$

in $\mathcal{C}^{\mathsf{op}}$, this means

$$A \xleftarrow{f} B \xleftarrow{g} C$$

in $\mathcal{C}$, so composition $g \circ f$ in $\mathcal{C}^{\mathsf{op}}$ is defined as $f \circ g$ in $\mathcal{C}$!

Consideration of opposite categories leads to a ***principle of duality***: a statement S is true about $\mathcal{C}$ if and only if its dual (i.e. the one obtained from S by reversing all the arrows) is true about $\mathcal{C}^{\mathsf{op}}$. For example,

A morphism f is monic in $\mathcal{C}^{\mathsf{op}}$ if and only if it is epic in $\mathcal{C}$.

Indeed, f is monic in $\mathcal{C}^{\mathsf{op}}$ iff for all $g, h : C \to B$ in $\mathcal{C}^{\mathsf{op}}$,

$$f \circ g = f \circ h \implies g = h\,,$$

iff for all $g, h : B \to C$ in $\mathcal{C}$,

$$g \circ f = h \circ f \implies g = h\,,$$

iff f is epic in $\mathcal{C}$. We say that monic and epic are *dual notions*.

Exercise 9. If P is a preorder, for example $(\mathbb{R}, \leq)$, describe P^{op} explicitly.

Subcategories

Another way to obtain new categories from old ones is by restricting their objects or arrows.

Definition 5 Let $\mathcal{C}$ be a category. Suppose that we are given collections

$$\mathsf{Ob}(\mathcal{D}) \subseteq \mathsf{Ob}(\mathcal{C})\,, \quad \forall A, B \in \mathsf{Ob}(\mathcal{D}).\ \mathcal{D}(A,B) \subseteq \mathcal{C}(A,B)$$

We say that $\mathcal{D}$ is a ***subcategory*** of $\mathcal{C}$ if

$$A \in \mathsf{Ob}(\mathcal{D}) \Rightarrow \mathsf{id}_A \in \mathcal{D}(A,A), \qquad f \in \mathcal{D}(A,B), g \in \mathcal{D}(B,C) \Rightarrow g \circ f \in \mathcal{D}(A,C)\,,$$

and hence $\mathcal{D}$ itself is a category. In particular, $\mathcal{D}$ is:

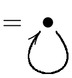
- A ***full*** subcategory of $\mathcal{C}$ if for any $A, B \in \mathsf{Ob}(\mathcal{D})$, $\mathcal{D}(A,B) = \mathcal{C}(A,B)$.
- A ***lluf*** subcategory of $\mathcal{C}$ if $\mathsf{Ob}(\mathcal{D}) = \mathsf{Ob}(\mathcal{C})$. ▲

For example, **Grp** is a full subcategory of **Mon** (by Exercise 2), and **Set** is a lluf subcategory of **Rel**.

Simple cats

We close this section with some very basic examples of categories.

- $\mathbb{1}$ is the category with one object and one arrow, that is

  $\mathbb{1} :=$ 

  where the arrow is necessarily $\mathsf{id}_\bullet$. Note that, although we say that $\mathbb{1}$ is *the* one-object/one-arrow category, there is by no means a *unique* such category. This is explained by the intuitively evident fact that any two such categories are isomorphic. (We will define what it means for categories to be isomorphic later).
- In two-object categories, there is the one with two arrows, $\mathbb{2} := \bullet \quad \bullet$, and also:

  $$\mathbb{2}_{\to} := \bullet \longrightarrow \bullet \quad , \quad \mathbb{2}_{\rightrightarrows} := \bullet \rightrightarrows \bullet \quad , \quad \mathbb{2}_{\rightleftarrows} := \bullet \rightleftarrows \bullet \quad \ldots$$

  Note that we have omitted identity arrows for economy. Categories with only identity arrows, like $\mathbb{1}$ and $\mathbb{2}$, are called *discrete categories*.

Exercise 10. How many categories $\mathcal{C}$ with $\mathsf{Ob}(\mathcal{C}) = \{\bullet\}$ are there? (Hint: what do such categories correspond to?)

1.1.6 Exercises

1. Consider the following properties of an arrow f in a category $\mathcal{C}$.

- f is *split monic* if for some g, $g \circ f$ is an identity arrow.
- f is *split epic* if for some g, $f \circ g$ is an identity arrow.

a) Prove that if f and g are arrows such that $g \circ f$ is monic, then f is monic.
b) Prove that, if f is split epic then it is epic.
c) Prove that, if f and $g \circ f$ are iso then g is iso.
d) Prove that, if f is monic and split epic then it is iso.
e) In the category **Mon** of monoids and monoid homomorphisms, consider the inclusion map

$$i : (\mathbb{N}, +, 0) \longrightarrow (\mathbb{Z}, +, 0)$$

of natural numbers into the integers. Show that this arrow is both monic and epic. Is it an iso?

The **Axiom of Choice** in Set Theory states that, if $\{X_i\}_{i \in I}$ is a family of non-empty sets, we can form a set $X = \{x_i \mid i \in I\}$ where $x_i \in X_i$ for all $i \in I$.

f) Show that in **Set** an arrow which is epic is split epic. Explain why this needs the Axiom of Choice.
g) Is it always the case that an arrow which is epic is split epic? Either prove that it is, or give a counter-example.

2. Give a description of partial orders as categories of a special kind.

1.2 Some Basic Constructions

We shall now look at a number of basic constructions which appear throughout mathematics, and which acquire their proper general form in the language of categories.

1.2.1 Initial and terminal objects

A first such example is that of initial and terminal objects. While apparently trivial, they are actually both important and useful, as we shall see in the sequel.

Definition 6 An object I in a category $\mathcal{C}$ is *initial* if, for every object A, *there exists a unique arrow* from I to A, which we write $\iota_A : I \to A$.
A *terminal* object in $\mathcal{C}$ is an object T such that, for every object A, *there exists a unique arrow* from A to T, which we write $\tau_A : A \to T$. ▲

Note that initial and terminal objects are dual notions: T is terminal in $\mathcal{C}$ iff it is initial in $\mathcal{C}^{\mathsf{op}}$. We sometimes write $\mathbf{1}$ for the terminal object and $\mathbf{0}$ for the initial one. Note also the assertions of *unique existence* in the definitions. This is one of the *leitmotifs* of category theory; we shall encounter it again in a conceptually deeper form in Chapter 5.

Let us examine initial and terminal objects in our standard example categories.

- In **Set**, the empty set is an initial object while any one-element set $\{\bullet\}$ is terminal.
- In **Pos**, the poset $(\varnothing, \varnothing)$ is an initial object while $(\{\bullet\}, \{(\bullet, \bullet)\})$ is terminal.
- In **Top**, the space $(\varnothing, \{\varnothing\})$ is an initial object while $(\{\bullet\}, \{\varnothing, \{\bullet\}\})$ is terminal.
- In $\mathbf{Vect}_k$, the one-element space $\{0\}$ is both initial and terminal.
- In a poset, seen as a category, an initial object is a least element, while a terminal object is a greatest element.

Exercise 11. Verify these claims. In each case, identify the canonical arrows.

Exercise 12. Identify the initial and terminal objects in **Rel**.

Exercise 13. Suppose that a monoid, viewed as a category, has either an initial or a terminal object. What must the monoid be?

We shall now establish a fundamental fact: initial and terminal objects are *unique up to (unique) isomorphism*. As we shall see, this is characteristic of all such "universal" definitions. For example, the apparent arbitrariness in the fact that any singleton set is a terminal object in **Set** is answered by the fact that what counts is the property of being terminal; and this suffices to ensure that any two concrete objects having this property must be isomorphic to each other.

The proof of the proposition, while elementary, is a first example of distinctively categorical reasoning.

Proposition 2. *If I and I' are initial objects in the category $\mathcal{C}$ then there exists a unique isomorphism $I \xrightarrow{\cong} I'$.*

Proof: Since I is initial and I' is an object of $\mathcal{C}$, there is a unique arrow $\iota_{I'} : I \longrightarrow I'$. We claim that $\iota_{I'}$ is an isomorphism.
Since I' is initial and I is an object in $\mathcal{C}$, there is an arrow $\iota'_I : I' \longrightarrow I$. Thus we obtain $\iota_{I'}; \iota'_I : I \longrightarrow I$, while we also have the identity morphism $\mathsf{id}_I : I \longrightarrow I$. But I is initial and therefore there exists a *unique* arrow from I to I, which means that $\iota_{I'}; \iota'_I = \mathsf{id}_I$. Similarly, $\iota'_I; \iota_{I'} = \mathsf{id}_{I'}$, so $\iota_{I'}$ is indeed an isomorphism. ■

Hence, initial objects are "unique up to (unique) isomorphism", and we can (and do) speak of *the* initial object (if any such exists). Similarly for terminal objects.

1.2.2 Products and Coproducts

Products

We now consider one of the most common constructions in mathematics: the formation of "direct products". Once again, rather than giving a case-by-case construction of direct products in each mathematical context we encounter, we can express once and for all a general notion of product, meaningful in any category — and such that, if a product exists, it is characterized uniquely up to unique isomorphism, just as for initial and terminal objects. Given a particular mathematical context, *i.e.* a category, we can then verify whether on not the product exists in that category. The concrete construction appropriate to the context will enter only into the proof of *existence*; all of the useful *properties* of the product follow from the general definition. Moreover, the categorical notion of product has a *normative* force; we can test whether a concrete construction works as intended by verifying that it satisfies the general definition.

In set theory, the cartesian product is defined in terms of the ordered pair:

$$X \times Y := \{(x, y) \mid x \in X \ \wedge \ y \in Y\}.$$

It turns out that ordered pairs can be *defined* in set theory, e.g. as

$$(x, y) := \{\{x, y\}, y\}.$$

Note that in no sense is such a definition canonical. The essential *properties* of ordered pairs are:

1. We can retrieve the first and second components x, y of the ordered pair (x, y), allowing *projection functions* to be defined:

$$\pi_1 : (x, y) \mapsto x, \qquad \pi_2 : (x, y) \mapsto y\,.$$

2. The information about first and second components completely determines the ordered pair:

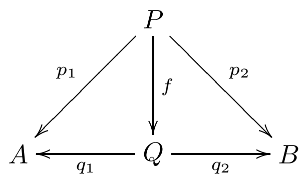
$$(x_1, x_2) = (y_1, y_2) \iff x_1 = y_1 \ \wedge \ x_2 = y_2.$$

The categorical definition expresses these properties in arrow-theoretic terms, meaningful in any category.

Definition 7 Let A, B be objects in a category $\mathcal{C}$. An A,B–pairing is a triple (P, p_1, p_2) where P is an object, $p_1 : P \to A$ and $p_2 : P \to B$. A morphism of A,B–pairings

$$f : (P, p_1, p_2) \longrightarrow (Q, q_1, q_2)$$

is a morphism $f : P \to Q$ in $\mathcal{C}$ such that $q_1 \circ f = p_1$ and $q_2 \circ f = p_2$, i.e. the following diagram commutes.



The A,B–pairings form a category $\mathbf{Pair}(A, B)$. We say that $(A \times B, \pi_1, \pi_2)$ is a ***product*** of A and B if it is *terminal* in $\mathbf{Pair}(A, B)$. ▲

Exercise 14. Verify that $\mathbf{Pair}(A, B)$ is a category.

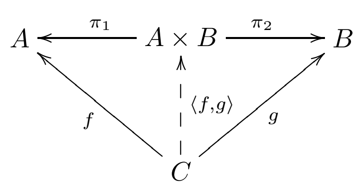
Note that products are specified by triples $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$, where π_i's are called *projections*. For economy (and if projections are obvious) we may say that $A \times B$ is the product of A and B. We say that $\mathcal{C}$ has (binary) *products* if each pair of objects A, B has a product in $\mathcal{C}$. A direct consequence of the definition, by Proposition 2, is that if products exist, they are unique up to (unique) isomorphism.

Unpacking the uniqueness condition from $\mathbf{Pair}(A, B)$ back to $\mathcal{C}$ we obtain a more concise definition of products which we use in practice.

Definition 8 (Equivalent definition of product) Let A, B be objects in a category $\mathcal{C}$. A product of A and B is an object $A \times B$ together with a pair of arrows $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that for every triple $A \xleftarrow{f} C \xrightarrow{g} B$ there exists a *unique* morphism

$$\langle f, g \rangle : C \longrightarrow A \times B$$

such that the following diagram commutes.



$$\begin{pmatrix} \pi_1 \circ \langle f, g \rangle = f \\ \pi_2 \circ \langle f, g \rangle = g \end{pmatrix}$$

▲

We call $\langle f, g \rangle$ the *pairing* of f and g.

Note that the above diagram features a *dashed arrow*. Our intention with such diagrams is always to express the following idea: if the undashed part of the diagram commutes, then *there exists a unique arrow* (the dashed one) such that the whole diagram commutes. In any case, we shall always spell out the intended statement explicitly.

We look at how this definition works in our standard example categories.

- In **Set**, products are the usual cartesian products.
- In **Pos**, products are cartesian products with the pointwise order.

- In **Top**, products are cartesian products with the product topology.
- In $\mathbf{Vect}_k$, products are direct sums.
- In a poset, seen as a category, products are *greatest lower bounds.*

Exercise 15. Verify these claims.

The following proposition shows that the uniqueness of the pairing arrow can be specified purely equationally.

Proposition 3. *For any triple* $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ *the following statements are equivalent.*

(I) For any triple $A \xleftarrow{f} C \xrightarrow{g} B$ *there exists a unique morphism* $\langle f, g\rangle : C \to A \times B$ *such that* $\pi_1 \circ \langle f, g\rangle = f$ *and* $\pi_2 \circ \langle f, g\rangle = g$.

(II) For any triple $A \xleftarrow{f} C \xrightarrow{g} B$ *there exists a morphism* $\langle f, g\rangle : C \to A \times B$ *such that* $\pi_1 \circ \langle f, g\rangle = f$ *and* $\pi_2 \circ \langle f, g\rangle = g$, *and moreover, for any* $h : C \to A \times B$, $h = \langle \pi_1 \circ h, \pi_2 \circ h\rangle$.

Proof: For (I)⇒(II), take any $h : C \longrightarrow A \times B$; we need to show $h = \langle \pi_1 \circ h, \pi_2 \circ h\rangle$. We have

$$A \xleftarrow{\pi_1 \circ h} C \xrightarrow{\pi_2 \circ h} B$$

and hence, by (I), there exists unique $k : C \longrightarrow A \times B$ such that

$$\pi_1 \circ k = \pi_1 \circ h \quad \wedge \quad \pi_2 \circ k = \pi_2 \circ h \qquad (*)$$

Note now that $(*)$ holds both for $k := h$ and $k := \langle \pi_1 \circ h, \pi_2 \circ h\rangle$, the latter because of (I). Hence, $h = \langle \pi_1 \circ h, \pi_2 \circ h\rangle$.

For (II)⇒(I), take any triple $A \xleftarrow{f} C \xrightarrow{g} B$. By (II), we have that there exists an arrow $\langle f, g\rangle : C \longrightarrow A \times B$ such that $\pi_1 \circ \langle f, g\rangle = f$ and $\pi_2 \circ \langle f, g\rangle = g$. We need to show it is the unique such. Let $k : C \longrightarrow A \times B$ s.t.

$$\pi_1 \circ k = f \quad \wedge \quad \pi_2 \circ k = g$$

Then, by (II),

$$k = \langle \pi_1 \circ k, \pi_2 \circ k\rangle = \langle f, g\rangle$$

as required. ■

In the following proposition we give some useful properties of products. First, let us introduce some notation for arrows: given $f_1 : A_1 \to B_1$, $f_2 : A_2 \to B_2$, define

$$f_1 \times f_2 := \langle f_1 \circ \pi_1, f_2 \circ \pi_2\rangle : A_1 \times A_2 \longrightarrow B_1 \times B_2.$$

Proposition 4. *For any* $f : A \to B$, $g : A \to C$, $h : A' \to A$, *and any* $p : B \to B'$, $q : C \to C'$,

- $\langle f, g\rangle \circ h = \langle f \circ h, g \circ h\rangle$,
- $(p \times q) \circ \langle f, g\rangle = \langle p \circ f, q \circ g\rangle$.

Proof: For the first claim we have:

$$\langle f, g\rangle \circ h = \langle \pi_1 \circ (\langle f, g\rangle \circ h), \pi_2 \circ (\langle f, g\rangle \circ h)\rangle = \langle f \circ h, g \circ h\rangle.$$

And for the second:

$$\begin{aligned}(p \times q) \circ \langle f, g\rangle &= \langle p \circ \pi_1, q \circ \pi_2\rangle \circ \langle f, g\rangle \\ &= \langle p \circ \pi_1 \circ \langle f, g\rangle, q \circ \pi_2 \circ \langle f, g\rangle\rangle \\ &= \langle p \circ f, q \circ g\rangle.\end{aligned}$$

■

General Products

The notion of products can be generalised to arbitrary arities as follows. A product for a family of objects $\{A_i\}_{i\in I}$ in a category $\mathcal{C}$ is an object P and morphisms

$$p_i : P \longrightarrow A_i \quad (i \in I)$$

such that, for all objects B and arrows

$$f_i : B \longrightarrow A_i \quad (i \in I)$$

there is a *unique* arrow

$$g : B \longrightarrow P$$

such that, for all $i \in I$, the following diagram commutes:

$$\begin{array}{ccc} B & \xrightarrow{\;g\;} & P \\ & {\scriptstyle f_i}\searrow \quad \swarrow{\scriptstyle p_i} & \\ & A_i & \end{array}$$

As before, if such a product exists, it is unique up to (unique) isomorphism. We write $P = \prod_{i\in I} A_i$ for the product object, and $g = \langle f_i \mid i \in I\rangle$ for the unique morphism in the definition.

Exercise 16. What is the product of the empty family?

Exercise 17. Show that if a category has binary and nullary products then it has all finite products.

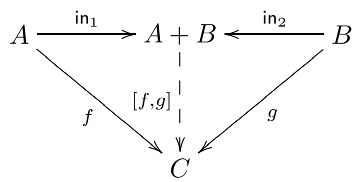
Coproducts

We now investigate the dual notion to products: namely coproducts. Formally, coproducts in C are just products in $\mathcal{C}^{\mathsf{op}}$, interpreted back in C . We spell out the definition.

Definition 9 Let A, B be objects in a category $\mathcal{C}$. A *coproduct* of A and B is an object $A + B$ together with a pair of arrows $A \xrightarrow{\mathsf{in}_1} A + B \xleftarrow{\mathsf{in}_2} B$ such that for every triple $A \xrightarrow{f} C \xleftarrow{g} B$ there exists a *unique* morphism

$$[f, g] : A + B \longrightarrow C$$

such that the following diagram commutes:



$$\begin{pmatrix} [f, g] \circ \mathsf{in}_1 = f \\ [f, g] \circ \mathsf{in}_2 = g \end{pmatrix}$$

▲

We call in_i's *injections* and $[f, g]$ a *copairing*. As with pairings, uniqueness of copairings can be specified by an equation:

$$\forall h : A + B \to C.\ h = [h \circ \mathsf{in}_1, h \circ \mathsf{in}_2]\ .$$

Coproducts in **Set**

This is given by *disjoint union* of sets, which can be defined concretely e.g. by

$$X + Y = \{1\} \times X \ \cup\ \{2\} \times Y.$$

We can define *injections*

$$X \xrightarrow{\mathsf{in}_1} X + Y \xleftarrow{\mathsf{in}_2} Y$$

$$\mathsf{in}_1(x) = (1, x) \qquad \mathsf{in}_2(y) = (2, y).$$

Also, given functions $f : X \longrightarrow Z$ and $g : Y \longrightarrow Z$, we can define

$$[f, g] : X + Y \longrightarrow Z$$

$$[f, g](1, x) = f(x) \qquad\qquad [f, g](2, y) = g(y).$$

Exercise 18. Check that this construction does yield coproducts in **Set**.

Note that this example suggests that coproducts allow for *definition by cases.*

Let us examine coproducts for some of our other standard examples.

- In **Pos**, disjoint unions (with the inherited orders) are coproducts.
- In **Top**, topological disjoint unions are coproducts.
- In $\mathbf{Vect}_k$, direct sums are coproducts.
- In a poset, *least upper bounds* are coproducts.

Exercise 19. Verify these claims.

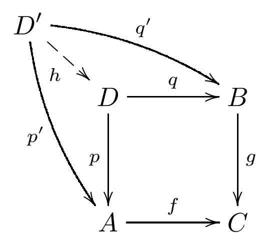
Exercise 20. Dually to products, express coproducts as initial objects of a category $\mathbf{Copair}(A, B)$ of A,B–copairings.

1.2.3 Pullbacks and Equalisers

We shall consider two further constructions of interest: *pullbacks* and *equalisers.*

Pullbacks

Definition 10 Consider a pair of morphisms $A \xrightarrow{f} C \xleftarrow{g} B$. The ***pull-back*** of f along g is a pair $A \xleftarrow{p} D \xrightarrow{q} B$ such that $f \circ p = g \circ q$ and, for any pair $A \xleftarrow{p'} D' \xrightarrow{q'} B$ such that $f \circ p' = g \circ q'$, there exists a unique $h : D' \to D$ such that $p' = p \circ h$ and $q' = q \circ h$. Diagrammatically,



▲

Example 1. • In **Set** the pullback of $A \xrightarrow{f} C \xleftarrow{g} B$ is defined as a *subset of the cartesian product*:

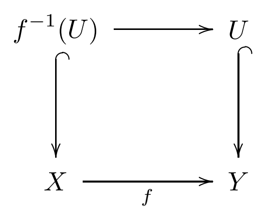
$$A \times_C B \ = \ \{(a, b) \in A \times B \mid f(a) = g(b)\}.$$

For example, given a category $\mathcal{C}$, with

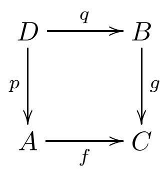
$$\mathsf{Ar}(\mathcal{C}) \xrightarrow{\mathsf{dom}} \mathsf{Ob}(\mathcal{C}) \xleftarrow{\mathsf{cod}} \mathsf{Ar}(\mathcal{C})\,.$$

Then the pullback of dom along cod is the set of *compable morphisms*, *i.e.* pair of morphisms (f, g) in $\mathcal{C}$ such that $g \circ f$ is well-defined.

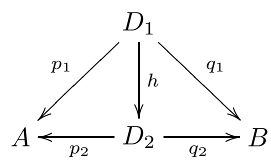
- In **Set** again, subsets (i.e. inclusion maps) pull back to subsets:



Just as for products, pullbacks can equivalently be described as terminal objects in suitable categories. Given a pair of morphisms $A \xrightarrow{f} C \xleftarrow{g} B$, we define an (f,g)–*cone* to be a pair (p,q) such that the following diagram commutes.



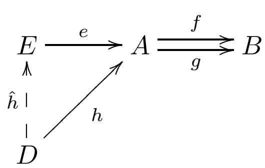
A morphism of (f,g)–cones $h : (D_1, p_1, q_1) \to (D_2, p_2, q_2)$ is a morphism $h : D_1 \to D_2$ such that the following diagram commutes.



We can thus form a category $\mathbf{Cone}(f,g)$. A pull-back of f along g, if it exists, is exactly a terminal object of $\mathbf{Cone}(f,g)$. Once again, this shows the uniqueness of pullbacks up to unique isomorphism.

Equalisers

Definition 11 Consider a pair of parallel arrows $A \overset{f}{\underset{g}{\rightrightarrows}} B$. An ***equaliser*** of (f,g) is an arrow $e : E \to A$ such that $f \circ e = g \circ e$ and, for any arrow $h : D \to A$ such that $f \circ h = g \circ h$ there is a unique $\hat{h} : D \to E$ so that $h = e \circ \hat{h}$. Diagrammatically,



▲

As for products, uniqueness of the arrow from D to E can be expressed equationally:

$$\forall k : D \to E.\ \widehat{e \circ k} = k\,.$$

Exercise 21. Why is $\widehat{e \circ k}$ well-defined for any $k : D \to E$? Prove that the above equation is equivalent to the uniqueness requirement.

Example 2. In **Set**, the equaliser of f, g is given by the inclusion

$$\{x \in A \mid f(x) = g(x)\} \hookrightarrow A\,.$$

This allows *equationally defined subsets* to be defined as equalisers. For example, consider the pair of maps $\mathbb{R}^2 \underset{g}{\overset{f}{\rightrightarrows}} \mathbb{R}$, where

$$f : (x, y) \mapsto x^2 + y^2, \qquad g : (x, y) \mapsto 1\,.$$

Then the equaliser is the unit circle as a subset of $\mathbb{R}^2$.

1.2.4 Limits and Colimits

The notions we have introduced so far are all special cases of a general notion of *limits* in categories, and the dual notion of *colimits*.

Limits	**Colimits**
Terminal Objects	Initial Objects
Products	Coproducts
Pullbacks	Pushouts
Equalisers	Coequalisers

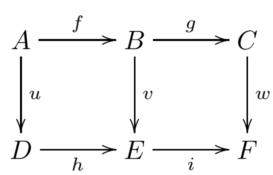
Table 1.1: Examples of limits and colimits

An important aspect of studying any kind of mathematical structure is to see what limits and colimits the category of such structures has. We shall return to these ideas shortly.

1.2.5 Exercises

1. Give an example of a category where some pair of objects lacks a product or coproduct.

2. (*Pullback lemma*) Consider the following commutative diagram.



Given that the right hand square $BCEF$ and the outer square $ACDF$ are pullbacks, prove that the left hand square $ABDE$ is a pullback.

3. Consider $A \xrightarrow{f} C \xleftarrow{g} B$ with pullback $A \xleftarrow{p} D \xrightarrow{q} B$. For each $A \xleftarrow{p'} D' \xrightarrow{q'} B'$ with $f \circ p' = g \circ q'$, let $\phi(p', q') : D' \to D$ be the arrow dictated by the pullback condition. Express uniqueness of $\phi(p', q')$ equationally.

1.3 Functors

Part of the "categorical philosophy" is:

Don't just look at the objects; take the morphisms into account too.

We can also apply this to categories!

1.3.1 Basics

A "morphism of categories" is a *functor*.

Definition 12 A ***functor*** $F : \mathcal{C} \to \mathcal{D}$ is given by:

- An object-map, assigning an object FA of $\mathcal{D}$ to every object A of $\mathcal{C}$.
- An arrow-map, assigning an arrow $Ff : FA \to FB$ of $\mathcal{D}$ to every arrow $f : A \to B$ of $\mathcal{C}$, in such a way that composition and identities are preserved:

$$F(g \circ f) = Fg \circ Ff\,, \qquad F\,\mathsf{id}_A = \mathsf{id}_{FA}\,.$$

▲

Note that we use the same symbol to denote the object- and arrow-maps. In practice, this never causes confusion.

The conditions on preservation of composition and identities are called *functoriality*.

Examples:

Example 3. Let $(P, \leq)$, $(Q, \leq)$ be preorders (seen as categories). A functor $F : (P, \leq) \longrightarrow (Q, \leq)$ is specified by an object-map, say $F : P \to Q$, and an appropriate arrow-map. The arrow-map corresponds to the condition

$$\forall p_1, p_2 \in P.\, p_1 \leq p_2 \implies F(p_1) \leq F(p_2)\,,$$

i.e. to monotonicity of F. Moreover, the functoriality conditions are trivial since in the codomain $(Q, \leq)$ all hom-sets are singletons.
Hence, a functor between preorders is just a monotone map.

Example 4. Let $(M, \cdot, 1)$, $(N, \cdot, 1)$ be monoids. A functor $F : (M, \cdot, 1) \longrightarrow (N, \cdot, 1)$ is specified by a trivial object map (monoids are categories with a single object) and an arrow-map, say $F : M \to N$. The functoriality conditions correspond to

$$\forall m_1, m_2 \in M.\, F(m_1 \cdot m_2) = F(m_1) \cdot F(m_2)\,, \qquad F(1) = 1\,,$$

i.e. to F being a monoid homomorphism.
Hence, a functor between monoids is just a monoid homomorphism.

Other examples are the following.

- Inclusion of a sub-category, $\mathcal{C} \hookrightarrow \mathcal{D}$, is a functor (by taking the identity map for object- and arrow-map).
- The *covariant* powerset functor $\mathcal{P}\mathbf{Set} \longrightarrow \mathbf{Set}$::

$$X \mapsto \mathcal{P}(X)\,, \qquad (f : X \longrightarrow Y) \mapsto \mathcal{P}(f) := S \mapsto \{f(x) \mid x \in S\}.$$

- $U : \mathbf{Mon} \longrightarrow \mathbf{Set}$ is the 'forgetful' or 'underlying' functor which sends a monoid to its set of elements, 'forgetting' the algebraic structure, and sends a homomorphism to the corresponding function between sets. There are similar forgetful functors for other categories of structured sets. Why are these trivial-looking functors useful? — We shall see!
- Group theory examples. The assignment of the commutator sub-group of a group extends to a functor from **Group** to **Group**; and the assignment of the quotient by this normal subgroup extends to a functor from **Group** to **AbGroup**. The assignment of the centralizer of a group does not!
- More sophisticated examples: e.g. *homology.* The basic idea of algebraic topology is that there are functorial assignments of algebraic objects (e.g. groups) to topological spaces, and variants of this idea ('(co)homology theories') are pervasive throughout modern pure mathematics.

Functors 'of several variables'

We can generalise the notion of a functor to a mapping from several domain categories to a codomain category. For this we need the following definition.

Definition 13 For categories $\mathcal{C}, \mathcal{D}$ define the ***product category*** $\mathcal{C} \times \mathcal{D}$ as follows. An object in $\mathcal{C} \times \mathcal{D}$ is a pair of objects from $\mathcal{C}$ and $\mathcal{D}$, and an arrow in $\mathcal{C} \times \mathcal{D}$ is a pair of arrows from $\mathcal{C}$ and $\mathcal{D}$. Identities and arrow composition are defined componentwise:

$$\mathsf{id}_{(A,B)} := (\mathsf{id}_A, \mathsf{id}_B)\,, \qquad (f, g) \circ (f', g') := (f \circ f', g \circ g')\,.$$

▲

A functor 'of two variables', with domains $\mathcal{C}$ and $\mathcal{D}$, to $\mathcal{E}$ is simply a functor:

$$F : \mathcal{C} \times \mathcal{D} \longrightarrow \mathcal{E}\,.$$

For example, there are evident projection functors

$$\mathcal{C} \longleftarrow \mathcal{C} \times \mathcal{D} \longrightarrow \mathcal{D}\,.$$

1.3.2 Further Examples

Set-valued functors

Many important constructions arise as functors $F : \mathcal{C} \rightarrow \mathbf{Set}$. For example:

- If G is a group, a functor $F : G \rightarrow \mathbf{Set}$ is an *action of* G *on a set.*

- If P is a poset representing time, a functor $F : P \to \mathbf{Set}$ is a notion of *set varying through time*. This is related to *Kripke semantics*, and to *forcing arguments* in set theory.
- Recall that $\mathbb{2}$ is the category $\bullet \rightrightarrows \bullet$. Then, functors $F : \mathbb{2} \to \mathbf{Set}$ correspond to *directed graphs* understood as in Definition 2, *i.e.* as structures (V, E, s, t), where V is a set of vertices, E is a set of edges, and $s, t : E \to V$ specify the source and target vertices for each edge.

Let us examine the first example in more detail. For a group $(G, \cdot, 1)$, a functor $F : G \to \mathbf{Set}$ is specified by a set X (where the unique object of G is mapped), and by an arrow-map sending each element m of G to an endofunction on X, say $m \bullet \_ : X \to X$. Then, functoriality amounts to the conditions

$$\forall m_1, m_2 \in G.\ F(m_1 \cdot m_2) = F(m_1) \circ F(m_2)\,, \qquad F(1) = \mathsf{id}_X\,,$$

that is, for all $m_1, m_2 \in G$ and all $x \in X$,

$$(m_1 \cdot m_2) \bullet x = m_1 \bullet m_2 \bullet x\,, \qquad 1 \bullet x = x\,.$$

We therefore see that F defines an action of G on X.

Exercise 22. Verify that functors $F : \mathbb{2} \to \mathbf{Set}$ correspond to directed graphs.

Example: Lists

Data-type constructors are functors. As a basic example, we consider lists. There is a functor

$$\mathsf{List} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

which takes a set X to the set of all finite lists (sequences) of elements of X. List is functorial: its action on morphisms (*i.e.* functions, *i.e.* (functional) programs) is given by *maplist*:

$$\frac{f : X \longrightarrow Y}{\mathsf{List}(f) : \mathsf{List}(X) \longrightarrow \mathsf{List}(Y)}$$

$$\mathsf{List}(f)[x_1, \ldots, x_n] = [f(x_1), \ldots, f(x_n)]$$

We can upgrade List to a functor $\mathsf{MList} : \mathbf{Set} \to \mathbf{Mon}$ by mapping each set X to the monoid $(\mathsf{List}(X), *, \epsilon)$ and $f : X \to Y$ to $\mathsf{List}(f)$, as above. The monoid operation $* : \mathsf{List}(X) \times \mathsf{List}(X) \to \mathsf{List}(X)$ is list concatenation, and ϵ is the empty list. We call $\mathsf{MList}(X)$ the ***free monoid*** over X. This terminology will be justified in Chapter 5.

Products as functors

If a category C has binary products, then there is automatically a functor

$$\_ \times \_ : \mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}$$

which takes each pair (A, B) to the product $A \times B$, and each (f, g) to

$$f \times g = \langle f \circ \pi_1, g \circ \pi_2 \rangle \,.$$

Functoriality is shown as follows, using proposition 4 and uniqueness of pairings in its equational form.

$$\begin{aligned}(f \times g) \circ (f' \times g') &= (f \times g) \circ \langle f' \circ \pi_1, g' \circ \pi_2 \rangle = \langle f \circ f' \circ \pi_1, g \circ g' \circ \pi_2 \rangle \\ &= (f \circ f') \times (g \circ g') \,, \\ \mathsf{id}_A \times \mathsf{id}_B &= \langle \mathsf{id}_A \circ \pi_1, \mathsf{id}_B \circ \pi_2 \rangle = \langle \pi_1 \circ \mathsf{id}_{A \times B}, \pi_2 \circ \mathsf{id}_{A \times B} \rangle = \mathsf{id}_{A \times B} \,.\end{aligned}$$

The category of categories

There is a category **Cat** whose objects are categories, and whose arrows are functors. Composition of functors is defined in the evident fashion. Note that if $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$ then, for $f : A \to B$ in $\mathcal{C}$,

$$G \circ F(f) := G(F(f)) : G(F(A)) \longrightarrow G(F(B))$$

so the types work out. A category of categories sounds (and is) circular, but in practice is harmless: one usually makes some size restriction on the categories, and then **Cat** will be too "big" to be an object of itself.

Note that product categories are products in **Cat**! For any pair of categories $\mathcal{C}, \mathcal{D}$, set

$$\mathcal{C} \xleftarrow{\boldsymbol{\pi}_1} \mathcal{C} \times \mathcal{D} \xrightarrow{\boldsymbol{\pi}_2} \mathcal{D}$$

where $\mathcal{C} \times \mathcal{D}$ the product category (defined previously) and $\boldsymbol{\pi}_i$'s the obvious projection functors. For any pair of functors $\mathcal{C} \xleftarrow{F} \mathcal{E} \xrightarrow{G} \mathcal{D}$, set

$$\langle F, G \rangle : \mathcal{E} \longrightarrow \mathcal{C} \times \mathcal{D} := A \mapsto (F(A), F(B)),\ f \mapsto (Ff, Gf) \,.$$

It is easy to see that $\langle F, G \rangle$ is indeed a functor. Moreover, satisfaction of the product diagram and uniqueness are shown exactly as in **Set**.

1.3.3 Contravariance

By definition, the arrow-map of a functor F is *covariant*: it preserves the direction of arrows, so if $f : A \to B$ then $Ff : FA \to FB$. A *contravariant* functor G does exactly the opposite: it reverses arrow-direction, so if $f : A \to B$ then $Gf : GB \to GA$. A concise way to express contravariance is as follows.

Definition 14 Let $\mathcal{C}, \mathcal{D}$ be categories. A ***contravariant*** functor G from $\mathcal{C}$ to $\mathcal{D}$ is a functor $G : \mathcal{C}^{\mathsf{op}} \to \mathcal{D}$. (Equivalently, a functor $G : \mathcal{C} \longrightarrow \mathcal{D}^{\mathsf{op}}$). ▲

Explicitly, a contravariant functor G is given by an assignment of:

- an object GA in $\mathcal{D}$ to every object A in $\mathcal{C}$,

- an arrow $Gf : GB \longrightarrow GA$ in $\mathcal{D}$ to every arrow $f : A \longrightarrow B$ in $\mathcal{C}$, such that (notice the change of order in composition):

$$G(g \circ f) = Gf \circ Gg\,, \qquad G\mathsf{id}_A = \mathsf{id}_{GA}.$$

Note that functors of several variables can be covariant in some variables and contravariant in others, e.g.

$$F : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \longrightarrow \mathcal{E}\,.$$

Examples of Contravariant Functors

- The contravariant powerset functor, $\mathcal{P}^{\mathsf{op}} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}$, is given by:

$$\mathcal{P}^{\mathsf{op}}(X) := \mathcal{P}(X)\,.$$
$$\mathcal{P}^{\mathsf{op}}(f : X \longrightarrow Y) : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X) := T \mapsto \{x \in X \mid f(x) \in T\}\,.$$

- The dual space functor on vector spaces:

$$(\_)^* : \mathbf{Vect}_k^{\mathsf{op}} \longrightarrow \mathbf{Vect}_k := V \mapsto V^*.$$

Note that these are both examples of the following idea: send an object A into functions from A into some fixed object. For example, the powerset can be written as $\mathcal{P}(X) = 2^X$, where we think of a subset in terms of its characteristic function.

Hom-functors

We now consider some fundamental examples of **Set**-valued functors, Given a category $\mathcal{C}$ and an object A of $\mathcal{C}$, two functors to **Set** can be defined:

- The covariant Hom-functor at A,

$$\mathcal{C}(A, \_) : \mathcal{C} \longrightarrow \mathbf{Set}\,,$$

which is given by (recall that each $\mathcal{C}(A, B)$ is a set):

$$\mathcal{C}(A, \_)(B) := \mathcal{C}(A, B)\,, \qquad \mathcal{C}(A, \_)(f : B \to C) := g \mapsto f \circ g\,.$$

We usually write $\mathcal{C}(A, \_)(f)$ as $\mathcal{C}(A, f)$. Functoriality reduces directly to the basic category axioms: associativity of composition and the unit laws for the identity.

- There is also a contravariant Hom-functor,

$$\mathcal{C}(\_, A) : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathbf{Set}\,,$$

given by:

$$\mathcal{C}(\_, A)(B) := \mathcal{C}(B, A)\,, \qquad \mathcal{C}(\_, A)(h : C \to B) := g \mapsto g \circ h\,.$$

Generalizing both of the above, we obtain a bivariant Hom-functor,

$$\mathcal{C}(\_, \_) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \longrightarrow \mathbf{Set}\,.$$

Exercise 23. Spell out the definition of $\mathcal{C}(\_, \_) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \longrightarrow \mathbf{Set}$. Verify carefully that it is a functor.

1.3.4 Properties of functors

Definition 15 A functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ is said to be:

- ***faithful*** if each map $F_{A,B} : \mathcal{C}(A, B) \longrightarrow \mathcal{D}(FA, FB)$ is injective.
- ***full*** if each map $F_{A,B} : \mathcal{C}(A, B) \longrightarrow \mathcal{D}(FA, FB)$ is surjective.
- an ***embedding*** if F is full, faithful, and injective on objects.
- an ***equivalence*** if F is full, faithful, and *essentially surjective*: i.e. for every object B of $\mathcal{D}$ there is an object A of $\mathcal{C}$ such that $F(A) \cong B$.
- An ***isomorphism*** if there is a functor $G : \mathcal{D} \longrightarrow \mathcal{C}$ such that

$$G \circ F = \mathsf{Id}_{\mathcal{C}}\,, \qquad F \circ G = \mathsf{Id}_{\mathcal{D}}\,.$$

  Note that this is just the usual notion of isomorphism applied to **Cat**. We say that categories $\mathcal{C}$ and $\mathcal{D}$ are isomorphic, $\mathcal{C} \cong \mathcal{D}$, if there is an isomorphism between them. ▲

Examples:

- The forgetful functor $U : \mathbf{Mon} \to \mathbf{Set}$ is faithful, but not full. For the latter, note that not all functions $f : M \to N$ yield an arrow $f : (M, \cdot, 1) \to (N, \cdot, 1)$. Similar properties hold for other forgetful functors.
- The free monoid functor $\mathsf{MList} : \mathbf{Set} \to \mathbf{Mon}$ is faithful, but not full.
- The product functor $\_ \times \_ : \mathcal{C} \times \mathcal{C} \longrightarrow \mathcal{C}$ is generally neither faithful nor full.
- There is an equivalence between $\mathbf{FDVect}_k$ the category of finite dimensional vector spaces over the field k, and $\mathbf{Mat}_k$, the category of matrices with entries in k. Note that these categories are very far from isomorphic! This example is elaborated in exercise 3.5(1).

Preservation and Reflection

Let P be a property of arrows. We say that a functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ *preserves* P if whenever f satisfies P, so does $F(f)$. We say that F *reflects* P if whenever $F(f)$ satisfies P, so does f. For example:

a. All functors preserve isomorphisms, split monics and split epics.
b. Faithful functors reflect monics and epics.
c. Full and faithful functors reflect isomorphisms.
d. Equivalences preserve monics and epics.
- The forgetful functor $U : \mathbf{Mon} \to \mathbf{Set}$ preserves products.

Let us show c; the rest are given as exercises below. So let $f : A \to B$ in $\mathcal{C}$ be such that Ff is an iso, that is it has an inverse $g' : FB \to FA$. Then, by fullness, there exists some $g : B \to A$ so that $g' = Fg$. Thus,

$$F(g \circ f) = Fg \circ Ff = g' \circ Ff = \mathsf{id}_{FA} = F(\mathsf{id}_A)\,.$$

By faithfulness we obtain $g \circ f = \mathsf{id}_A$. Similarly, $f \circ g = \mathsf{id}_B$ and therefore f is an isomorphism.

Exercise 24. Show items a, b and d above.

Exercise 25. Show the following.

- Functors do not in general reflect monics or epics.
- Faithful functors do not in general reflect isomorphisms.
- Full and faithful functors do not in general preserve monics or epics.

1.3.5 Exercises

1. Consider the category $\mathbf{FDVect}_{\mathbb{R}}$ of finite dimensional vector spaces over $\mathbb{R}$, and $\mathbf{Mat}_{\mathbb{R}}$ of matrices over $\mathbb{R}$. Concretely, $\mathbf{Mat}_{\mathbb{R}}$ is defined as follows:

$$Ob(\mathbf{Mat}_{\mathbb{R}}) := \mathbb{N}\,,$$
$$\mathbf{Mat}_{\mathbb{R}}(n,m) := \{M \mid M \text{ is an } n \times m \text{ matrix with entries in } \mathbb{R}\}\,.$$

Thus, objects are natural numbers, and arrows $n \to m$ are $n \times m$ real matrices. Composition is matrix multiplication, and the identity on $\mathbf{n}$ is the $n \times n$ diagonal matrix.
Now let $F : \mathbf{Mat}_{\mathbb{R}} \longrightarrow \mathbf{FDVect}_{\mathbb{R}}$ be the functor taking each n to the vector space $\mathbb{R}^n$ and each $M : n \to m$ to the linear function

$$FM : \mathbb{R}^n \longrightarrow \mathbb{R}^m := (x_1, ..., x_n) \mapsto [x_1, ..., x_n]M$$

with the $1 \times m$ matrix $[x_1, ..., x_n]M$ considered as a vector in $\mathbb{R}^m$. Show that F is full, faithful and essentially surjective, and hence that $\mathbf{FDVect}_{\mathbb{R}}$ and $\mathbf{Mat}_{\mathbb{R}}$ are equivalent categories. Are they isomorphic?
2. Let $\mathcal{C}$ be a category with binary products such that, for each pair of objects A, B,

$$\mathcal{C}(A,B) \neq \emptyset. \qquad (*)$$

Show that the product functor $F : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is faithful.
Would F still be faithful in the absence of condition $(*)$?

1.4 Natural transformations

> *"Categories were only introduced to allow functors to be defined; functors were only introduced to allow natural transformations to be defined."*

Just as categories have morphisms between them, namely functors, so functors have morphisms between them too — *natural transformations*.

1.4.1 Basics

Definition 16 Let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A natural transformation

$$t : F \longrightarrow G$$

is a family of morphisms in $\mathcal{D}$ indexed by objects A of $\mathcal{C}$,

$$\{\ t_A : FA \longrightarrow GA\ \}_{A \in Ob(\mathcal{C})}$$

such that, for all $f : A \to B$, the following diagram commutes.

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ {\scriptstyle t_A}\downarrow & & \downarrow{\scriptstyle t_B} \\ GA & \xrightarrow[Gf]{} & GB \end{array}$$

This condition is known as *naturality*.
If each t_A is an isomorphism, we say that t is a *natural isomorphism*:

$$t : F \xrightarrow{\cong} G\,.$$

▲

Examples:

- Let Id be the identity functor on **Set**, and $\times \circ \langle \mathsf{Id}, \mathsf{Id} \rangle$ be the functor taking each set X to $X \times X$ and each function f to $f \times f$. Then there is a natural transformation $d : \mathsf{Id} \longrightarrow \times \circ \langle \mathsf{Id}, \mathsf{Id} \rangle$ given by:

  $$d_X : X \longrightarrow X \times X := x \mapsto (x, x)\,.$$

  Naturality amounts to asserting that, for any function $f : X \to Y$, the following diagram commutes:

  $$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle d_X}\downarrow & & \downarrow{\scriptstyle d_Y} \\ X \times X & \xrightarrow[f \times f]{} & Y \times Y \end{array}$$

  We call d the *diagonal* transformation on **Set**. In fact, it is the *only* natural transformation between these functors.
- The diagonal transformation can be defined for any category $\mathcal{C}$ with binary products by setting, for each object A in $\mathcal{C}$,

  $$d_A : A \longrightarrow A \times A := \langle \mathsf{id}_A, \mathsf{id}_A \rangle\,.$$

Projections also yield natural transformations. For example the arrows

$$\pi_{1(A,B)} : A \times B \longrightarrow A := \pi_1$$

specify a natural transformation $\pi_1 : \times \to \boldsymbol{\pi}_1$. Note that $\times, \boldsymbol{\pi}_1 : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ are the functors for product and first projection respectively.

- Let $\mathcal{C}$ be a category with terminal object T, and let $K_T : \mathcal{C} \to \mathcal{C}$ be the functor mapping all objects to T and all arrows to id_T. Then the canonical arrows

$$\tau_A : A \longrightarrow T$$

specify a natural transformation $\tau : \mathsf{Id} \to K_T$ (where Id the identity functor on $\mathcal{C}$).

- Recall the functor $\mathsf{List} : \mathbf{Set} \to \mathbf{Set}$ which takes a set X to the set of finite lists with elements in X. We can define (amongst others) the following natural transformations,

$$\mathsf{reverse} : \mathsf{List} \longrightarrow \mathsf{List}\,, \quad \mathsf{unit} : \mathsf{Id} \longrightarrow \mathsf{List}\,, \quad \mathsf{flatten} : \mathsf{List} \circ \mathsf{List} \longrightarrow \mathsf{List}\,,$$

by setting, for each set X,

$$\begin{aligned} \mathsf{reverse}_X &: \mathsf{List}(X) \longrightarrow \mathsf{List}(X) := [x_1, \ldots, x_n] \mapsto [x_n, \ldots, x_1]\,, \\ \mathsf{unit}_X &: X \longrightarrow \mathsf{List}(X) := x \mapsto [x]\,, \\ \mathsf{flatten}_X &: \mathsf{List}(\mathsf{List}(X)) \longrightarrow \mathsf{List}(X) \\ &:= [\,[x_1^1, \ldots, x_{n_1}^1], \ldots, [x_1^k, \ldots, x_{n_k}^k]\,] \mapsto [x_1^1, \ldots\ldots, x_{n_k}^k]\,. \end{aligned}$$

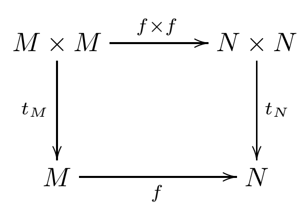
- Consider the following functor.

$$\times \circ \langle U, U \rangle : \mathbf{Mon} \longrightarrow \mathbf{Set} = (M, \cdot, 1) \mapsto M \times M,\ f \mapsto f \times f\,.$$

Then, the monoid operation yields a natural transformation $t : \times \circ \langle U, U \rangle \to U$ defined by:

$$t_{(M,\cdot,1)} : M \times M \longrightarrow M := (m, m') \mapsto m \cdot m'\,.$$

Naturality corresponds to asserting that, for any $f : (M, \cdot, 1) \to (N, \cdot, 1)$, the following diagram commutes,



that is, for any $m_1, m_2 \in M$, $f(m_1) \cdot f(m_2) = f(m_1 \cdot m_2)$.

- If V is a finite dimensional vector space, then V is isomorphic to both its first dual V^* and to its second dual V^{**}.
  However, while it is naturally isomorphic to its second dual, there is no natural isomorphism to the first dual. This was actually the original example which motivated Eilenberg and Mac Lane to define the concept of natural transformation; here naturality captures *basis independence*.

Exercise 26. Verify naturality of diagonal transformations, projections and terminals for a category $\mathcal{C}$ with finite products.

Exercise 27. Prove that the diagonal is the only natural transformation $\mathsf{Id} \longrightarrow \times \circ \langle \mathsf{Id}, \mathsf{Id} \rangle$ on **Set**. Similarly, prove that the first projection is the only natural transformation $\times \to \boldsymbol{\pi}_1$ on **Set**.

1.4.2 Further examples

Natural isomorphisms for products

Let $\mathcal{C}$ be a category $\mathcal{C}$ with finite products, i.e. binary products and a terminal object $\mathbf{1}$. Then we have the following canonical natural isomorphisms.

$$\begin{aligned} a_{A,B,C} &: A \times (B \times C) \xrightarrow{\cong} (A \times B) \times C\,, \\ s_{A,B} &: A \times B \xrightarrow{\cong} B \times A\,, \\ l_A &: \mathbf{1} \times A \xrightarrow{\cong} A\,, \\ r_A &: A \times \mathbf{1} \xrightarrow{\cong} A\,. \end{aligned}$$

The first two isomorphisms are meant to assert that the product is *associative* and *symmetric*, and the last two that $\mathbf{1}$ is its *unit*. In later sections we will see that these conditions form part of the definition of *symmetric monoidal categories*.

These natural isomorphisms are defined explicitly by:

$$\begin{aligned} a_{A,B,C} &:= \langle \langle \pi_1, \pi_1 \circ \pi_2 \rangle, \pi_2 \circ \pi_2 \rangle\,, \\ s_{A,B} &:= \langle \pi_2, \pi_1 \rangle\,, \\ l_A &:= \pi_2\,, \\ r_A &:= \pi_1\,. \end{aligned}$$

Since natural isomorphisms are a *self-dual* notion, similar natural isomorphisms can be defined if $\mathcal{C}$ has binary coproducts and an initial object.

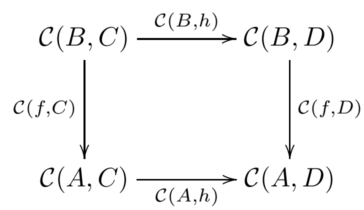
Exercise 28. Verify that these families of arrows are natural isomorphisms.

Natural transformations between Hom-functors

Let $f : A \to B$ in a category $\mathcal{C}$. Then this induces a natural transformation

$$\mathcal{C}(f, \_) : \mathcal{C}(B, \_) \longrightarrow \mathcal{C}(A, \_)\,,$$
$$\mathcal{C}(f, \_)_C : \mathcal{C}(B, C) \longrightarrow \mathcal{C}(A, C) := (g : B \to C) \mapsto (g \circ f : A \to C)\,.$$

Note that $\mathcal{C}(f, \_)_C$ is the same as $\mathcal{C}(f, C)$, the result of applying the contravariant functor $\mathcal{C}(\_, C)$ to f. Hence, naturality amounts to asserting that, for each $h : C \to D$, the following diagram commutes.



Starting from a $g : B \to C$, we compute:

$$\mathcal{C}(A, h)(\mathcal{C}(f, C)(g)) = h \circ (g \circ f) = (h \circ g) \circ f = \mathcal{C}(f, D)(\mathcal{C}(B, h)(g))\,.$$

The natural transformation $\mathcal{C}(\_, f) : \mathcal{C}(\_, A) \to \mathcal{C}(\_, B)$ is defined similarly.

Exercise 29. Define the natural transformation $\mathcal{C}(\_, f)$ and verify its naturality.

There is a remarkable result, the ***Yoneda Lemma***, which says that *every* natural transformation between Hom-functors comes from a (unique) arrow in $\mathcal{C}$ in the fashion described above.

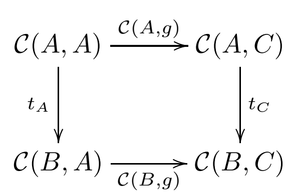
Lemma 1. *Let A, B be objects in a category $\mathcal{C}$. For each natural transformation $t : \mathcal{C}(A, \_) \to \mathcal{C}(B, \_)$, there is a unique arrow $f : B \to A$ such that*

$$t = \mathcal{C}(f, \_)\,.$$

Proof: Take any such A, B and t and let

$$f : B \longrightarrow A := t_A(\mathsf{id}_A)\,.$$

We want to show that $t = \mathcal{C}(f, \_)$. For any object C and any arrow $g : A \to C$, naturality of t means that the following commutes.



Starting from id_A we have that:

$$t_C(\mathcal{C}(A,g)(\mathsf{id}_A)) = \mathcal{C}(B,g)(t_A(\mathsf{id}_A)) , \quad \text{i.e.} \quad t_C(g) = g \circ f .$$

Hence, noting that $\mathcal{C}(f,C)(g) = g \circ f$, we obtain $t = \mathcal{C}(f,\_)$.
For uniqueness we have that, for any $f, f' : B \to A$, if $\mathcal{C}(f,\_) = \mathcal{C}(f',\_)$ then

$$f = \mathsf{id}_A \circ f = \mathcal{C}(f,A)(\mathsf{id}_A) = \mathcal{C}(f',A)(\mathsf{id}_A) = \mathsf{id}_A \circ f' = f'.$$

■

Exercise 30. Prove a similar result for contravariant hom-functors.

Alternative definition of equivalence

Another way of defining equivalence of categories is as follows.

Definition 17 We say that categories $\mathcal{C}$ and $\mathcal{D}$ are ***equivalent***, $\mathcal{C} \simeq \mathcal{D}$, if there are functors $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{D} \to \mathcal{C}$ and natural isomorphisms

$$G \circ F \cong \mathsf{Id}_{\mathcal{C}} , \quad F \circ G \cong \mathsf{Id}_{\mathcal{D}} .$$

▲

1.4.3 Functor Categories

Suppose we have functors $F, G, H : \mathcal{C} \longrightarrow \mathcal{D}$ and natural transformations

$$t : F \longrightarrow G , \quad u : G \longrightarrow H .$$

Then we can compose these natural transformations, yielding $u \circ t : F \to H$:

$$(u \circ t)_A := F(A) \xrightarrow{t_A} G(A) \xrightarrow{u_A} H(A).$$

Composition is associative, and has as identity the natural transformation

$$I_F : F \longrightarrow F := \{ \ (I_F)_A := \mathsf{id}_A : F(A) \longrightarrow F(A) \ \}_A .$$

These observations lead us to the following.

Definition 18 For categories $\mathcal{C}, \mathcal{D}$ define the ***functor category*** $\mathsf{Func}(\mathcal{C}, \mathcal{D})$ by taking:

- Objects: functors $F : \mathcal{C} \longrightarrow \mathcal{D}$.
- Arrows: natural transformations $t : F \longrightarrow G$.

Composition and identities are given as above. ▲

Remark 19 We see that in the category **Cat** of categories and functors, each hom-set $\mathbf{Cat}(\mathcal{C}, \mathcal{D})$ *itself has the structure of a category.* In fact, **Cat** is the basic example of a "2-category", i.e. of a category where hom-sets are themselves categories.

Note that a natural isomorphism is precisely an isomorphism in the functor category. Let us proceed to some examples of functor categories.

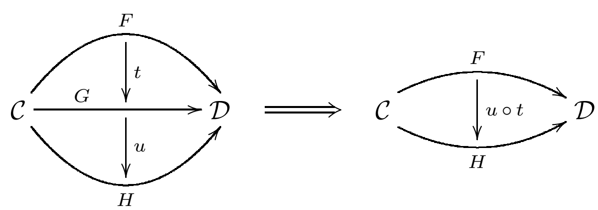
- Recall that, for any group G, functors from G to **Set** are G-actions on sets. Then, $\mathsf{Func}(G, \mathbf{Set})$ is the category of G-actions on sets and *equivariant functions*: $f : X \to Y$ such that $f(m \bullet x) = m \bullet f(x)$.
- $\mathsf{Func}(2, \mathbf{Set})$: Graphs and graph homomorphisms.
- If $F, G : P \to Q$ are monotone maps between posets, then $t : F \to G$ means that

$$\forall x \in P.\ Fx \leq Gx\,.$$

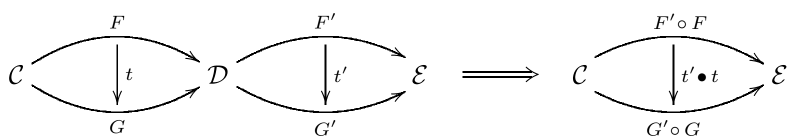
  Note that in this case naturality is trivial (hom-sets are singletons in Q).

Exercise 31. Verify the above descriptions of $\mathsf{Func}(G, \mathbf{Set})$ and $\mathsf{Func}(2, \mathbf{Set})$.

Remark 20 The composition of natural transformations defined above is called *vertical composition*. The reason for this terminology is depicted below.



As expected, there is also a *horizontal composition*, which is given as follows.



1.4.4 Exercises

1. Show that the two definitions of equivalence of categories, namely
   a) $\mathcal{C}$ and $\mathcal{D}$ are equivalent if there is an equivalence $F : \mathcal{C} \to \mathcal{D}$ (definition 15),
   b) $\mathcal{C}$ and $\mathcal{D}$ are equivalent if there are $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{D} \to \mathcal{C}$, and isomorphisms $F \circ G \cong \mathsf{Id}_{\mathcal{D}}$, $G \circ F \cong \mathsf{Id}_{\mathcal{C}}$ (definition 17),

   are: equivalent! Note that this will need the Axiom of Choice.

2. Define a relation on objects in a category $\mathcal{C}$ by: $A \cong B$ iff A and B are isomorphic.
   a) Show that this relation is an equivalence relation.

   Define a *skeleton* of $\mathcal{C}$ to be the (full) subcategory obtained by choosing one object from each equivalence class of $\cong$ (note that this involves choices, and is not uniquely defined).

b) Show that $\mathcal{C}$ is equivalent to any skeleton.
c) Show that any two skeletons of $\mathcal{C}$ are isomorphic.
d) Give an example of a category whose objects form a proper class, but whose skeleton is finite.

3. Given a category $\mathcal{C}$, we can define a functor

$$y : \mathcal{C} \longrightarrow \mathsf{Func}(\mathcal{C}^{\mathsf{op}}, \mathbf{Set}) := A \mapsto \mathcal{C}(\_, A), \quad f \mapsto \mathcal{C}(\_, f).$$

Prove carefully that this is indeed a functor. Use exercise 30 to conclude that y is full and faithful. Prove that it is also injective on objects, and hence an embedding. It is known as the *Yoneda embedding*.

4. Define the vertical composition $u \bullet t$ of natural transformations explicitly. Prove that it is associative.

1.5 Universality and Adjoints

There is a fundamental triad of categorical notions:

Functoriality, Naturality, Universality.

We have studied the first two notions explicitly. We have also seen many *examples* of universal definitions, notably the various notions of limits and colimits considered in section 2. It is now time to consider universality in general; the proper formulation of this fundamental and pervasive notion is one of the major achievements of basic category theory.

Universality arises when we are interested in finding *canonical solutions* to problems of construction: that is, we are interested, not just in the *existence* of a solution, but in its *canonicity*. This canonicity should guarantee uniqueness, in the sense we have become familiar with; a canonical solution should be *unique up to (unique) isomorphism*.

The notion of canonicity has a simple interpretation in the case of posets, as an *extremal solution*: one that is the least or the greatest among all solutions. Such an extremal solution is obviously unique. For example, consider the problem of finding a lower bound of a pair of elements A, B in a poset P: a *greatest lower bound* of A and B is an extremal solution to this problem. As we have seen, this is the specialization to posets of the problem of constructing a product:

- $\leadsto$ A product of A, B in a poset is an element C such that $C \leq A$ and $C \leq B$, (C is a lower bound);
- $\leadsto$ and for any other other solution C', i.e. C' such that $C' \leq A$ and $C' \leq B$, we have $C' \leq C$. (C is a greatest lower bound.)

Because the ideas of universality and adjunctions have an appealingly simple form in posets, which is, moreover, useful in its own right, we will develop the ideas in that special case first, as a prelude to the general discussion for categories.

1.5.1 Adjunctions for posets

Suppose $g : Q \to P$ is a monotone map between posets. Given $x \in P$, a ***g*-approximation of** x (from above) is an element $y \in Q$ such that $x \leq g(y)$. A ***best g*-approximation of** x is an element $y \in Q$ such that

$$x \leq g(y) \;\wedge\; \forall z \in Q.\, (\, x \leq g(z) \implies y \leq z\,)\,.$$

If a best g-approximation exists then it is clearly unique.

Discussion

It is worth clarifying the notion of best g-approximation. If y is a best g-approximation to x, then in particular, by monotonicity of g, $g(y)$ is the least element of the set of all $g(z)$ where $z \in Q$ and $x \leq g(z)$. However, the property of being a best approximation is much *stronger* than the mere existence of a least element of this set. We are asking for y *itself* to be the least, *in* Q, among all elements z such that $x \leq g(z)$. Thus even if g is *surjective*, so that for every x there is a $y \in Q$ such that $g(y) = x$, there need not exist a *best* g-approximation to x. This is exactly the issue of having a *canonical choice* of solution.

Exercise 32. Given a example of a surjective monotone map $g : Q \to P$ and an element $x \in P$ such that there is no best g-approximation to x in Q.

If such a best g-approximation $f(x)$ exists for all $x \in P$ then we have a function $f : P \to Q$ such that, for all $x \in P$, $z \in Q$:

$$x \leq g(z) \iff f(x) \leq z\,. \tag{1.1}$$

We say that f is the ***left adjoint*** of g, and g is the ***right adjoint*** of f. It is immediate from the definitions that the left adjoint of g, if it exists, is uniquely determined by g.

Proposition 5. *If such a function f exists, then it is monotone. Moreover,*

$$\mathsf{id}_P \leq g \circ f\,, \qquad f \circ g \leq \mathsf{id}_Q\,, \qquad f \circ g \circ f = f\,, \qquad g \circ f \circ g = g\,.$$

Proof: If we take $z = f(x)$ in equation (1.1), then since $f(x) \leq f(x)$, $x \leq g \circ f(x)$. Similarly, taking $x = g(z)$ we obtain $f \circ g(z) \leq z$. Now, the ordering on functions $h, k : P \longrightarrow Q$ is the *pointwise order*:

$$h \leq k \iff \forall x \in P.\, h(x) \leq k(x).$$

This gives the first two equations.

Now if $x \leq_P x'$, then $x \leq x' \leq g \circ f(x')$, so $f(x')$ is a g-approximation of x, and hence $f(x) \leq f(x')$. Thus f is monotone.

Finally, using the fact that composition is monotone with respect to the pointwise order on functions, and the first two equations:

$$g = \mathsf{id}_P \circ g \leq g \circ f \circ g \leq g \circ \mathsf{id}_Q = g,$$

and hence $g = g \circ f \circ g$. The other equation is proved similarly. ■

Examples:

- Consider the inclusion map

  $$i : \mathbb{Z} \hookrightarrow \mathbb{R}\,.$$

  This has both a left adjoint f^L and a right adjoint f^R, where $f^L, f^R : \mathbb{R} \to \mathbb{Z}$. For all $z \in \mathbb{Z}$, $r \in \mathbb{R}$:

  $$z \leq f^R(r) \iff i(z) \leq r\,, \qquad f^L(r) \leq z \iff r \leq i(z)\,.$$

  We see from these defining properties that the right adjoint maps a real r to the *greatest integer below it* (the extremal solution to finding an integer below a given real). This is the standard *floor function*.
  Similarly, the left adjoint maps a real to the least integer above it yielding the *ceiling function*. Thus:

  $$f^R(r) = \lfloor r \rfloor\,, \qquad f^L(r) = \lceil r \rceil\,.$$

- Consider a relation $R \subseteq X \times Y$. R induces a function:

  $$f_R : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y) := S \mapsto \{y \in Y \mid \exists x \in S.\, xRy\}\,.$$

  This has a right adjoint $[R] : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X)$:

  $$S \subseteq [R]T \iff f_R(S) \subseteq T\,.$$

  The definition of $[R]$ which satisfies this condition is:

  $$[R]T := \{x \in X \mid \forall y \in Y.\, xRy \;\Rightarrow\; y \in T\}\,.$$

  If we consider a set of *worlds* W with an *accessibility relation* $R \subseteq W \times W$ as in Kripke semantics for modal logic, we see that $[R]$ gives the usual Kripke semantics for the modal operator $\Box$, seen as a propositional operator mapping the set of worlds satisfied by a formula ϕ to the set of worlds satisfied by $\Box\phi$.
  On the other hand, if we think of the relation R as the denotation of a (possibly non-deterministic) program, and T as a predicate on *states*, then $[R]T$ is exactly the *weakest precondition* $\mathbf{wp}(R, T)$. In *Dynamic Logic*, the two settings are combined, and we can write expressions such as $[R]T$ directly, where T will be (the denotation of) some formula, and R the relation corresponding to a program.

- Consider a function $f : X \to Y$. This induces a function:

$$f^{-1} : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X) := T \mapsto \{x \in X \mid f(x) \in T\}\,.$$

This function f^{-1} has both a left adjoint $\exists(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$, and a right adjoint $\forall(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$. For all $S \subseteq X$, $T \subseteq Y$:

$$\exists(f)(S) \subseteq T \iff S \subseteq f^{-1}(T)\,, \qquad f^{-1}(T) \subseteq S \iff T \subseteq \forall(f)(S)\,.$$

How can we define $\forall(f)$ and $\exists(f)$ explicitly so as to fulfil these defining conditions? – As follows:

$$\begin{aligned} \exists(f)(S) &:= \{y \in Y \mid \exists x \in X.\, f(x) = y \;\wedge\; x \in S\}\,, \\ \forall(f)(S) &:= \{y \in Y \mid \forall x \in X.\, f(x) = y \;\Rightarrow\; x \in S\}\,. \end{aligned}$$

If $R \subseteq X \times Y$, which we write in logical notation as $R(x, y)$, and we take the projection function $\pi_1 : X \times Y \longrightarrow X$, then:

$$\forall(\pi_1)(R) \equiv \forall y.\, R(x, y)\,, \qquad \exists(\pi_1)(R) \equiv \exists y.\, R(x, y)\,.$$

This extends to an algebraic form of the usual Tarski model-theoretic semantics for first-order logic, in which:

Quantifiers are Adjoints.

Couniversality

We can dualize the discussion, so that starting with a monotone map $f : P \longrightarrow Q$ and $y \in Q$, we can ask for the best P-approximation to y from below: $x \in P$ such that $f(x) \leq y$, and for all $z \in P$:

$$f(z) \leq y \iff z \leq x.$$

If such a best approximation $g(y)$ exists for all $y \in Q$, we obtain a monotone map $g : Q \longrightarrow P$ such that g is right adjoint to f. From the symmetry of the definition, if it clear that:

$$f \text{ is the left adjoint of } g \iff g \text{ is the right adjoint of } f$$

and each determines the other uniquely.

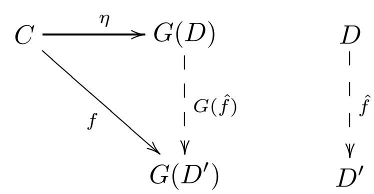
1.5.2 Universal Arrows and Adjoints

Our discussion of best approximations for posets is lifted to general categories as follows.

Definition 21 Let $G : \mathcal{D} \to \mathcal{C}$ be a functor, and C an object of $\mathcal{C}$. A ***universal arrow from*** C ***to*** G is a pair (D, η) where D is an object of $\mathcal{D}$ and

$$\eta : C \longrightarrow G(D)\,,$$

such that, for any object D' of $\mathcal{D}$ and morphism $f : C \to G(D')$, *there exists a unique* morphism $\hat{f} : D \to D'$ in $\mathcal{D}$ such that $f = G(\hat{f}) \circ \eta$.
Diagrammatically:



▲

As in previous cases, uniqueness can be given a purely equational specification:

$$\forall h : D \longrightarrow D'.\ \widehat{G(h) \circ \eta} = h\,. \tag{1.2}$$

Exercise 33. Show that if (D, η) and (D', η') are universal arrows from C to G then there is a unique isomorphism $D \cong D'$.

Exercise 34. Check that the equational specification of uniqueness (1.2) is valid.

Examples:

- Take $U : \mathbf{Mon} \to \mathbf{Set}$. Given a set X, the universal arrow is

  $$\eta_X : X \longrightarrow U(\mathsf{MList}(X)) := x \mapsto [x]\,.$$

  Indeed, for any monoid $(M, \cdot, 1)$ and any function $f : X \to M$, set

  $$\hat{f} : \mathsf{MList}(X) \longrightarrow (M, \cdot, 1) := [x_1, \ldots, x_n] \mapsto f(x_1) \cdot \cdots \cdot f(x_n)\,.$$

  It is easy to see that $\hat{f}$ is a monoid homomorphism, and that $U(\hat{f}) \circ \eta_X = f$. Moreover, for uniqueness we have that, for any $h : \mathsf{MList}(X) \to (M, \cdot, 1)$,

  $$\begin{aligned} \widehat{U(h) \circ \eta_X} &= \widehat{x \mapsto h([x])} = [x_1, \ldots, x_n] \mapsto h([x_1]) \cdot \cdots \cdot h([x_n]) \\ &= [x_1, \ldots, x_n] \mapsto h([x_1] * \cdots * [x_n]) \\ &= [x_1, \ldots, x_n] \mapsto h([x_1, \ldots, x_n]) = h\,. \end{aligned}$$

- Let $K : \mathcal{C} \to \mathbb{1}$ be the unique functor to the one-object/one-arrow category. A universal arrow from the object of $\mathbb{1}$ to K corresponds to an initial object in $\mathcal{C}$.
  Indeed, such a universal arrow is given by an object I of $\mathcal{C}$ (and a trivial arrow in $\mathbb{1}$), such that for any A in $\mathcal{C}$ (and relevant arrow in $\mathbb{1}$) there exists a unique arrow from I to A (such that a trivial condition holds).

- Consider the functor $\langle \mathsf{Id}_{\mathcal{C}}, \mathsf{Id}_{\mathcal{C}} \rangle : \mathcal{C} \to \mathcal{C} \times \mathcal{C}$, taking each object A to (A, A) and each arrow f to (f, f). A universal arrow from an object (A, B) of $\mathcal{C} \times \mathcal{C}$ to $\langle \mathsf{Id}_{\mathcal{C}}, \mathsf{Id}_{\mathcal{C}} \rangle$ corresponds to a coproduct of A and B.

Exercise 35. Verify the description of coproducts as universal arrows.

As in the case of posets, a related notion to universal arrows is that of *adjunction.*

Definition 22 Let $\mathcal{C}, \mathcal{D}$ be categories. An ***adjunction*** from $\mathcal{C}$ to $\mathcal{D}$ is a triple (F, G, θ), where F and G are functors

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

and θ is a family of bijections

$$\theta_{A,B} : \mathcal{C}(A, G(B)) \xrightarrow{\cong} \mathcal{D}(F(A), B)\,,$$

for each $A \in Ob(\mathcal{C})$ and $B \in Ob(\mathcal{D})$, **natural** in A and B.
We say that F is ***left adjoint*** to G, and G is ***right adjoint*** to F. ▲

Note that θ should be understood as the "witnessed" form — i.e. arrows instead of mere relations — of the defining condition for adjunctions in the case of posets:

$$x \leq g(y) \iff f(x) \leq y.$$

This is often displayed as a two-way 'inference rule':

$$\frac{C \longrightarrow GC}{FC \longrightarrow D}$$

Naturality of θ is expressed as follows: for any $f : A \to G(B)$ and any $g : A' \to A$, $h : B \to B'$,

$$\theta_{A',B}(f \circ g) = \theta_{A,B}(f) \circ F(g)\,,$$
$$\theta_{A,B'}(G(h) \circ f) = h \circ \theta_{A,B}(f)\,.$$

In one line:

$$\theta_{A',B'}(G(h) \circ f \circ g) = h \circ \theta_{A,B}(f) \circ F(g)\,.$$

Diagrammatically:

$$\begin{array}{ccccc} \mathcal{C}(A,GB') & \xleftarrow{\mathcal{C}(A,Gh)} & \mathcal{C}(A,GB) & \xrightarrow{\mathcal{C}(g,GB)} & \mathcal{C}(A',GB) \\ {\scriptstyle \theta_{A,B'}}\downarrow & & {\scriptstyle \theta_{A,B}}\downarrow & & \downarrow{\scriptstyle \theta_{A',B}} \\ \mathcal{D}(FA,B') & \xleftarrow[\mathcal{D}(FA,h)]{} & \mathcal{D}(FA,B) & \xrightarrow[\mathcal{D}(Fg,B)]{} & \mathcal{D}(FA',B) \end{array}
\qquad
\begin{array}{ccc} \mathcal{C}(A,GB) & \xrightarrow{\mathcal{C}(g,Gh)} & \mathcal{C}(A',GB') \\ {\scriptstyle \theta_{A,B}}\downarrow & & \downarrow{\scriptstyle \theta_{A',B'}} \\ \mathcal{D}(FA,B) & \xrightarrow[\mathcal{D}(Fg,h)]{} & \mathcal{D}(FA',B') \end{array}$$

Thus, θ is in fact a natural isomorphism

$$\theta : \mathcal{C}(\_, G(\_)) \longrightarrow \mathcal{D}(F(\_), \_)\,,$$

where $\mathcal{C}(\_, G(\_)) : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathbf{Set}$ is the result of composing the bivariant hom-functor $\mathcal{C}(\_, \_)$ with $\mathsf{Id}_{\mathcal{C}^{\mathsf{op}}} \times G$, and $\mathcal{D}(F(\_), \_)$ is similar.

In the next two propositions we show that universal arrows and adjunctions are equivalent notions.

Proposition 6 (Universals define adjunctions). *Let $G : \mathcal{D} \to \mathcal{C}$. If for every object C of $\mathcal{C}$ there exists a universal arrow $\eta_C : C \to G(F(C))$, then:*

1. *F uniquely extends to a functor $F : \mathcal{C} \to \mathcal{D}$ such that $\eta : \mathsf{Id}_{\mathcal{C}} \to G \circ F$ is a natural transformation.*
2. *F is uniquely determined by G (up to unique natural isomorphism), and vice versa.*
3. *For each pair of objects C of $\mathcal{C}$ and D of $\mathcal{D}$, there is a natural bijection:*

$$\theta_{C,D} : \mathcal{C}(C, G(D)) \cong \mathcal{D}(F(C), D)\,.$$

Proof: For 1, we extend F to a functor as follows. Given $f : C \to C'$ in $\mathcal{C}$, we consider the composition

$$\eta_{C'} \circ f : C \longrightarrow GFC'.$$

By the universal property of η_C, there exists a unique arrow $Ff : FC \to FC'$ such that the following diagram commutes.

$$\begin{array}{ccc} C & \xrightarrow{\eta_C} & GFC \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle GFf} \\ C' & \xrightarrow[\eta_{C'}]{} & GFC' \end{array}$$

Note that the above is the naturality diagram for η on C, hence the arrow-map thus defined for F is the unique candidate that makes η a natural transformation.
It remains to verify the functoriality of F. To show that F preserves composition, consider $g : C' \to C''$. We have the following commutative diagram,

$$\begin{array}{ccccc} C & \xrightarrow{f} & C' & \xrightarrow{g} & C'' \\ {\scriptstyle \eta_C}\downarrow & & {\scriptstyle \eta_{C'}}\downarrow & & \downarrow{\scriptstyle \eta_{C''}} \\ GFC & \xrightarrow[GFf]{} & GFC' & \xrightarrow[GFg]{} & GFC'' \end{array}$$

from which it follows that

$$G(Fg \circ Ff) \circ \eta_C = GFg \circ GFf \circ \eta_C = \eta_{C''} \circ g \circ f\,.$$

By definition, $F(g \circ f)$ is the unique arrow $h : FC \to FC'$ such that $Gh \circ \eta_C = \eta_{C''} \circ (g \circ f)$. By uniqueness, it follows that $Fg \circ Ff = h = F(g \circ f)$. The verification that F preserves identities is similar.
For 2, we have that each FC is determined uniquely up to unique isomorphism, by the universal property, and once the object part of F is fixed, the arrow part is uniquely determined.
For 3, we need to define a natural isomorphism $\theta_{C,D} : \mathcal{C}(C, G(D)) \cong \mathcal{D}(F(C), D)$. Given $f : C \to GD$, $\theta_{C,D}(f)$ is defined to be the unique arrow $FC \to D$ such that the following commutes, as dictated by universality.

$$\begin{array}{ccc} C & \xrightarrow{\eta_C} & GFC \\ & {\scriptstyle f}\searrow & \downarrow {\scriptstyle G(\theta_{C,D}(f))} \\ & & GD \end{array}$$

Suppose that $\theta_{C,D}(f) = \theta_{C,D}(g)$. Then

$$f = G(\theta_{C,D}(f)) \circ \eta_C = G(\theta_{C,D}(g)) \circ \eta_C = g\,.$$

Thus $\theta_{C,D}$ is injective. Moreover, given $h : FC \to D$, by the equational formulation of uniqueness we have:

$$h = \theta_{C,D}(Gh \circ \eta_C)\,.$$

Thus $\theta_{C,D}$ is surjective. We are left to show naturality, i.e. that the following diagram commutes, for all $h : C' \to C$ and $g : D \to D'$.

$$\begin{array}{ccc} \mathcal{C}(C, G(D)) & \xrightarrow{\mathcal{C}(h, G(g))} & \mathcal{C}(C', G(D')) \\ {\scriptstyle \theta_{C,D}}\downarrow & & \downarrow{\scriptstyle \theta_{C',D'}} \\ \mathcal{D}(F(C), D) & \xrightarrow[\mathcal{D}(F(h), g)]{} & \mathcal{D}(F(C'), D') \end{array}$$

We chase around the diagram, starting from $f : C \to G(D)$.

$$\mathcal{D}(F(h), g) \circ \theta_{C,D}(f) = g \circ \theta_{C,D}(f) \circ F(h)$$

$$\theta_{C',D'} \circ \mathcal{C}(h, G(g))(f) = \theta_{C',D'}(G(g) \circ f \circ h)$$

Now:

$$\begin{aligned} g \circ \theta_{C,D}(f) \circ F(h) &= \theta_{C',D'}(G(g \circ \theta_{C,D}(f) \circ F(h)) \circ \eta_{C'}) && \text{by (1.2)} \\ &= \theta_{C',D'}(Gg \circ G(\theta_{C,D}(f)) \circ GF(h) \circ \eta_{C'}) && \text{functoriality of } G \\ &= \theta_{C',D'}(Gg \circ G(\theta_{C,D}(f)) \circ \eta_C \circ h) && \text{naturality of } \eta \\ &= \theta_{C',D'}(Gg \circ f \circ h) && \text{by (1.2).} \end{aligned}$$

■

Proposition 7 (Adjunctions define universals). *Let $G : \mathcal{D} \to \mathcal{C}$ be a functor, $D \in Ob(\mathcal{D})$ and $C \in Ob(\mathcal{C})$. If, for any $D' \in Ob(\mathcal{D})$, there is a bijection*

$$\phi_{D'} : \mathcal{D}(D, D') \cong \mathcal{C}(C, G(D'))$$

natural in D' then there is a universal arrow $\eta : C \to G(D)$.

Proof: Take $\eta : C \to G(D) := \phi_D(\mathsf{id}_D)$ and, for any $g : C \to G(D')$, take $\hat{g} : D \to D' := \phi_{D'}^{-1}(g)$.
We have that

$$G(\hat{g}) \circ \eta = G(\hat{g}) \circ \phi_D(\mathsf{id}_D) \overset{\text{nat}}{=} \phi_{D'}(\hat{g}) = g\,.$$

Moreover, for any $h : D \to D'$,

$$\phi_{D'}^{-1}(Gh \circ \eta) = \phi_{D'}^{-1}(Gh \circ \phi_D(\mathsf{id}_D)) \overset{\text{nat}}{=} \phi_{D'}^{-1}(\phi_D(h)) = h\,,$$

as required. ■

Equivalence of Universal and Adjoints

Thus we see that the following two situations are equivalent,

- We are given a functor $G : \mathcal{D} \to \mathcal{C}$, and for each object C of $\mathcal{C}$ a universal arrow from C to G.
- We are given functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$, and a natural bijection

$$\theta_{C,D} : \mathcal{C}(C, G(D)) \cong \mathcal{D}(F(C), D)\,.$$

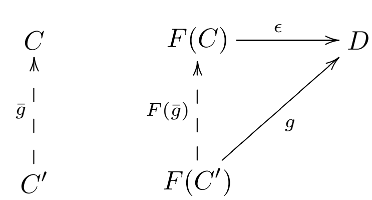
in the sense that each determines the other uniquely.

Couniversal Arrows

Let $F : \mathcal{C} \to \mathcal{D}$ be a functor, and D an object of $\mathcal{D}$. A *couniversal arrow from F to D* is an object C of $\mathcal{C}$ and a morphism

$$\epsilon : F(C) \longrightarrow D$$

such that, for every object C' of $\mathcal{C}$ and morphism $g : F(C') \to D$, there exists a unique morphism $\bar{g} : C' \longrightarrow C$ in $\mathcal{C}$ such that $g = \epsilon \circ F(\bar{g})$.
Diagrammatically:



By exactly similar (but dual) reasoning to the previous propositions, an adjunction implies the existence of couniversal arrows, and the existence of the latter implies the existence of the adjunction. Hence,

$$\text{Universality} \equiv \text{Adjunctions} \equiv \text{Couniversality}\,.$$

Some examples of couniversal arrows:

- A terminal object in a category $\mathcal{C}$ is a couniversal arrow from the unique functor $K : \mathcal{C} \to \mathbb{1}$ to the unique object in $\mathbb{1}$.
- Let A, B be objects of $\mathcal{C}$. A product of A and B is a couniversal arrow from $\langle \mathsf{Id}_{\mathcal{C}}, \mathsf{Id}_{\mathcal{C}} \rangle : \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ to (A, B).

1.5.3 Limits and colimits

In the previous paragraph we described products $A \times B$ as couniversal arrows from the *diagonal functor* $\Delta : \mathcal{C} \to \mathcal{C} \times \mathcal{C}$ to (A, B). Δ is the functor assigning (A, A) to each object A, and (f, f) to each arrow f. Noting that $\mathcal{C} \times \mathcal{C} = \mathcal{C}^{2}$, where $\mathcal{C}^{2}$ is a *functor category*, this suggests an important generalization.

Definition 23 Let $\mathcal{C}$ be a category and $\mathcal{I}$ be another category, thought of as an 'index category'. A *diagram of shape* $\mathcal{I}$ in $\mathcal{C}$ is just a functor $F : \mathcal{I} \to \mathcal{C}$. Consider the functor category $\mathcal{C}^{\mathcal{I}}$ with objects the functors from $\mathcal{I}$ to $\mathcal{C}$, and natural transformations as morphisms. There is a *diagonal functor*

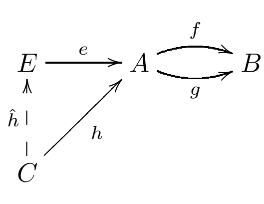
$$\Delta : \mathcal{C} \longrightarrow \mathcal{C}^{\mathcal{I}},$$

taking each object C of $\mathcal{C}$ to the constant functor $K_C : \mathcal{I} \to \mathcal{C}$, which maps every object of $\mathcal{I}$ to C. A ***limit*** for the diagram F is a couniversal arrow from Δ to F. ▲

This concept of limit subsumes products (including infinite products), pullbacks, inverse limits, etc.
For example, take $\mathcal{I} := \mathbb{2}_{\rightrightarrows}$ (we have seen this before: $\mathbb{2}_{\rightrightarrows} = \bullet \rightrightarrows \bullet$). A functor F from $\mathcal{I}$ to $\mathcal{C}$ corresponds to a diagram:

$$A \overset{f}{\underset{g}{\rightrightarrows}} B$$

A couniversal arrow from Δ to F corresponds to the following situation,



i.e. to an equaliser!

By dualizing limits we obtain *colimits*. Some important examples are coproducts, coequalisers, pushouts and ω-colimits.

Exercise 36. Verify that pullbacks are limits by taking:

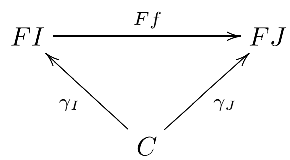
$$\mathcal{I} := \bullet \longrightarrow \bullet \longleftarrow \bullet$$

Limits as terminal objects

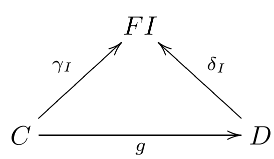
Consider $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{I}}$ and $F : \mathcal{I} \to \mathcal{C}$. A *cone to* F is an object C of $\mathcal{C}$ and family of arrows γ,

$$\{ \ \gamma_I : C \longrightarrow FI \ \}_{I \in Ob(\mathcal{I})} \, ,$$

such that, for any $f : I \to J$, the following triangle commutes.



Thus a cone is exactly a natural transformation $\gamma : \Delta C \longrightarrow F$. A morphism of cones ('mediating morphism') $(C, \gamma) \longrightarrow (D, \delta)$ is an arrow $g : C \to D$ such that each of the following triangles commutes.



We obtain a category $\mathbf{Cone}(F)$ whose objects are cones to F and whose arrows are mediating morphisms. Then, a limit of F is a terminal object in $\mathbf{Cone}(F)$.

1.5.4 Exponentials

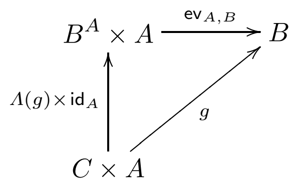
In $\mathbf{Set}$, given sets A, B, we can form the set of functions $B^A := \mathbf{Set}(A, B)$, *which is again a set, i.e. an object of* ***Set***. This closure of $\mathbf{Set}$ under forming 'function spaces' is one of its most important properties.

How can we axiomatise this situation? Once again, rather than asking what the elements of a function space *are*, we ask instead what we can *do* with them operationally. The answer is simple: apply functions to their arguments. That is, there is a map

$$\mathsf{ev}_{A,B} : B^A \times A \longrightarrow B \ \text{ such that } \ \mathsf{ev}_{A,B}(f, a) = f(a) \, .$$

We can think of the function as a ‘black box’: we can feed it inputs and observe the outputs.

Evaluation has the following couniversal property. For any $g : C \times A \to B$, there is a unique map $\Lambda(g) : C \to B^A$ such that the following diagram commutes.



commutes. In **Set**, this is defined by:

$$\Lambda(g)(c) : A \longrightarrow B := a \mapsto g(c, a)\,.$$

This process of transforming a function of two arguments into a function-valued function of one argument is known as *currying*, after H. B. Curry. It is an algebraic form of λ-*abstraction.*

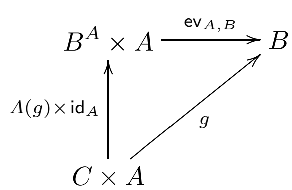
We are now led to the general definition of exponentials. Note that, for each object A of a category $\mathcal{C}$ with products, we can define a functor

$$\_ \times A : \mathcal{C} \longrightarrow \mathcal{C}\,.$$

Definition 24 Let $\mathcal{C}$ be a category with binary products. We say that $\mathcal{C}$ ***has exponentials*** if for all objects A and B of $\mathcal{C}$ there is a couniversal arrow from $\_ \times A$ to B, i.e. an object B^A of $\mathcal{C}$ and a morphism

$$\mathsf{ev}_{A,B} : B^A \times A \longrightarrow B$$

with the couniversal property: for every $g : C \times A \to B$, there is a unique morphism $\Lambda(g) : C \to B^A$ such that the following diagram commutes.



▲

Equivalently, $\mathcal{C}$ has exponentials if, for every object A, the functor $\_ \times A$ has a right adjoint, that is there exists a functor $\_^A : \mathcal{C} \to \mathcal{C}$ and a bijection

$$\Lambda_{B,C} : \mathcal{C}(C \times A, B) \xrightarrow{\cong} \mathcal{C}(C, B^A)$$

natural in B, C.

Notation 25 The notation B^A for exponential objects is standard in the category theory literature. For our purposes, however, it will be more convenient to write $A \Rightarrow B$.

Exponentials bring us to another fundamental notion, this time for understanding functional types, models of λ-calculus, and the structure of proofs.

Definition 26 A category with a terminal object, products and exponentials is called a ***Cartesian Closed Category (CCC)***.

For example, **Set** is a CCC. Another class of examples are *Boolean algebras*, seen as categories:

- Products are given by conjunctions $A \wedge B$. We define exponentials as *implications*:
$$A \Rightarrow B \ := \ \neg A \vee B \,.$$
- Evaluation is just *Modus Ponens*,
$$(A \Rightarrow B) \wedge A \ \leq \ B$$
while couniversality is the *Deduction Theorem*,
$$C \wedge A \ \leq \ B \iff C \ \leq \ A \Rightarrow B \,.$$

1.5.5 Exercises

1. Suppose that $U : \mathcal{C} \to \mathcal{D}$ has a left adjoint F_1, and $V : \mathcal{D} \to \mathcal{E}$ has a left adjoint F_2. Show that $V \circ U : \mathcal{C} \to \mathcal{E}$ has a left adjoint.

2. A *sup-lattice* is a poset P in which every subset $S \subseteq P$ has a supremum (least upper bound) $\bigvee S$. Let P, Q be sup-lattices, and $f : P \to Q$ be a monotone map.

a) Show that if f has a right adjoint then f preserves least upper bounds:
$$f(\bigvee S) = \bigvee \{ f(x) \mid x \in S \} \,.$$

b) Show that if f preserves least upper bounds then it has a right adjoint g, given by:
$$g(y) = \bigvee \{ x \in P \mid f(x) \leq y \} \,.$$

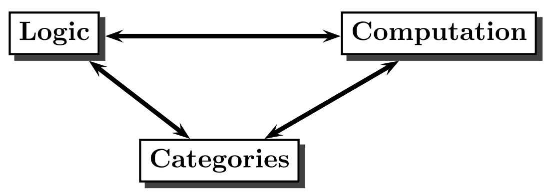
c) Dualise to get a necessary and sufficient condition for the existence of left adjoints.

3. Let $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{D} \to \mathcal{C}$ be functors such that F is left adjoint to G, with natural bijection $\theta_{C,D} : \mathcal{C}(C, GD) \xrightarrow{\cong} \mathcal{D}(FC, D)$. Show that there is a natural transformation $\varepsilon : F \circ G \to \mathsf{Id}_{\mathcal{D}}$, the ***counit*** of the adjunction. Describe this counit explicitly in the case where the right adjoint is the forgetful functor $U : \mathbf{Mon} \to \mathbf{Set}$.

4. Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be functors, and assume F is left adjoint to G with natural bijection θ.
   a) Show that F preserves epimorphisms.
   b) Show that F is faithful if and only if, for every object A of $\mathcal{C}$, $\eta_A : A \to GF(A)$ is monic.
   c) Show that if, for each object A of $\mathcal{C}$, there is a morphism $s_A : GF(A) \longrightarrow A$ such that $\eta_A \circ s_A = \mathsf{id}_{GF(A)}$ then F is full.

1.6 The Curry-Howard isomorphism

We shall now study a beautiful three-way connection between logic, computation and categories:



This connection has been known since the 1970's, and is widely used in Computer Science — it is also beginning to be used in Quantum Informatics! It is the upper link (Logic – Computation) that is usually attributed to Haskell B. Curry and William A. Howard, although the idea can be traced back to Brouwer, Heyting and Kolmogorov, and their interpretation of intuitionistic logic. The link to Categories is mainly due to the pioneering work of Joachim Lambek.

1.6.1 Logic

Suppose we ask ourselves the question: *What is Logic about?* There are two main kinds of answer: one focuses on *Truth*, and the other on *Proof*. We focus on the latter, that is on:

What follows from what

Traditional introductions to logic focus on Hilbert-style proof systems, that is on generating the set of *theorems* of a system from a set of *axioms* by applying rules of inference (e.g. Modus Ponens).

A key step in logic took place in the 1930's, with the advent of *Gentzen-style systems*. Instead of focusing on theorems, we look more generally and symmetrically at *What follows from what*: in these systems the primary focus is on *proofs from assumptions*.

Definition 27 Consider the fragment of propositional logic with logical connectives $\wedge$ and $\supset$. The assertion that a formula A can be proved from assumptions $A_1, ..., A_n$ is expressed by a ***sequent***:

$$A_1, \ldots, A_n \vdash A$$

We use Γ, Δ to range over finite sets of formulas, and write Γ, A for $\Gamma \cup \{A\}$. Proofs are built using the following proof rules; the resulting proof system is called *the* ***Natural Deduction system*** *for* $\wedge$,$\supset$.

Identity	**Conjunction**	**Implication**
$\dfrac{}{\Gamma, A \vdash A}\ \mathsf{Id}$	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\mathsf{intro}$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ \supset\mathsf{intro}$
	$\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge\mathsf{elim}_1$	$\dfrac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \supset\mathsf{elim}$
	$\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge\mathsf{elim}_2$	

▲

For example, we have the following proof for $\supset$-transitivity.

$$\dfrac{\dfrac{}{A \supset B, B \supset C, A \vdash B \supset C}\ \mathsf{Id} \quad \dfrac{\dfrac{}{A \supset B, B \supset C, A \vdash A \supset B}\ \mathsf{Id} \quad \dfrac{}{A \supset B, B \supset C, A \vdash A \supset A}\ \mathsf{Id}}{A \supset B, B \supset C, A \vdash B}\ \supset\mathsf{E}}{\dfrac{A \supset B, B \supset C, A \vdash C}{A \supset B, B \supset C \vdash A \supset C}\ \supset\mathsf{I}}\ \supset\mathsf{E}$$

noindent An important feature of Natural Deduction is the systematic pattern it exhibits in the structure of the inference rules. For each connective $\Box$, there are *introduction rules*, which show how formulas $A \Box B$ can be derived, and *elimination rules*, which show how such formulas can be used to derive other formulas.

Admissibility

We say that a proof rule

$$\dfrac{\Gamma_1 \vdash A_1 \quad \cdots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}$$

is ***admissible*** in Natural Deduction if, whenever there are proofs of $\Gamma_i \vdash A_i$ then there is also a proof of $\Delta \vdash B$. For example, the following *Cut rule* is admissible.

$$\dfrac{\Gamma \vdash A \quad \Gamma, A \vdash B}{\Gamma \vdash B}\ \mathsf{Cut}$$

Exercise 37. Use induction on the size of proofs to show the following:

1. The Weakening rule:
$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}$$
2. The Cut rule.

Our focus will be on ***Structural Proof Theory***, that is studying the 'space of formal proofs' as a mathematical structure in its own right, rather than focussing only on

$$\text{Provability} \longleftrightarrow \text{Truth}$$

(i.e. the usual notions of 'soundness and completeness'). One motivation for this approach comes from trying to understand and use the *computational content* of proofs, epitomised in the 'Curry-Howard correspondence'.

1.6.2 Computation

Our starting point in computation is the pure calculus of functions called the λ-*calculus*.

Definition 28 (λ-calculus) λ-calculus ***terms*** are constructed from a countably infinite set of ***variables*** by applying *applications* and λ-*abstractions*.

$$\begin{aligned}
&\text{VA} \ni x, y, z, \ldots \\
&\text{TE} \ni t, u \ ::= \ x \ | \ t\,u \ | \ \lambda x.\, t
\end{aligned}$$

▲

The computational content of the calculus is exhibited in the following examples.

$\lambda x.\, x + 1$	successor function
$\lambda x.\, x$	identity function
$\lambda f.\, \lambda x.\, f x$	application
$\lambda f.\, \lambda x.\, f(f x)$	double application
$\lambda f.\, \lambda g.\, \lambda x.\, g(f(x))$	composition and application

Note that the first example is not part of our formal syntax; it presupposes some *encoding* of numerals and successors.

The notation $\lambda x.t$ is meant to serve the purpose of expressing formally

the function that returns t on input x.

Thus λ is a *binder*, that is it binds the variable x in the 'function' $\lambda x.t$, in the same way that e.g. $\int$ binds x in $\int f(x)\, dx$. This means that there should not be a difference between $\lambda x.t$ and $\lambda x'.t'$, where t' is the result of swapping x with some *fresh* variable x' inside t. For example, the terms

$$\lambda x.x \text{ and } \lambda x'.x'$$

should be ‘equivalent’, as they both stand for the identity function. We formalize this by stipulating that

Terms are identified up to α-equivalence

where we say that two terms are α-equivalent iff they differ by some permutation of variables appearing in *binding positions.*

We now proceed to give a formal definition of α-equivalence. The definition is given in two steps, as follows.

Definition 29 We define *variable-swapping* on terms recursively as follows.

$$\begin{aligned}
(y\ x) \bullet z &:= \begin{cases} y & \text{if } z = x \\ x & \text{if } z = y \\ z & \text{otherwise} \end{cases} \\
(y\ x) \bullet u\,v &:= ((y\ x) \bullet u)((y\ x) \bullet v) \\
(y\ x) \bullet \lambda z.u &:= \lambda((y\ x) \bullet z).((y\ x) \bullet u)
\end{aligned}$$

Then, ***α-equivalence***, $=_\alpha$, is the relation on terms defined inductively by:[3]

- $x =_\alpha x$,
- $M\,N =_\alpha M'\,N'$ if $M =_\alpha M'$ and $N =_\alpha N'$,
- $\lambda x.M =_\alpha \lambda x'.M'$ if, for all y not appearing in $M\,M'$, $(y\ x) \bullet M =_\alpha (y\ x') \bullet M'$. ▲

The *free variables* of a term are those that are not bound by any λ; they can be seen as the *assumptions* of the term. Formally, the set of ***free variables*** of a term t, $\mathsf{fv}(t)$, is given by:

$$\begin{aligned}
\mathsf{fv}(y) &:= \{y\} \\
\mathsf{fv}(u\,v) &:= \mathsf{fv}(u) \cup \mathsf{fv}(v) \\
\mathsf{fv}(\lambda z.u) &:= \mathsf{fv}(u) \setminus \{z\}\,.
\end{aligned}$$

Exercise 38. Show that, for all terms t, t', if $t =_\alpha t'$ then $\mathsf{fv}(t) = \mathsf{fv}(t')$. Moreover, show that, for any $x, y \notin \mathsf{fv}(t)$, $t =_\alpha (x\ y) \bullet t$. Hence infer that, for any $y \notin \mathsf{fv}(t)$, $\lambda x.\, t =_\alpha \lambda y.\, (y\ x) \bullet t$.

[3] In fact, the definition can be given be replacing the last clause by any of the following:

- …if, for some y not appearing in $M\,M'$, $(y\ x) \bullet M =_\alpha (y\ x') \bullet M'$.
- …if, for all y not appearing free in $M\,M'$, $(y\ x) \bullet M =_\alpha (y\ x') \bullet M'$.
- …if, for some y not appearing free in $M\,M'$, $(y\ x) \bullet M =_\alpha (y\ x') \bullet M'$.

Since λ-abstractions stand for functions, an application of a λ-abstraction on another term should result to a *substitution* of the latter inside the body of the abstraction.

Definition 30 Define the ***substitution*** of a term t for a variable x inside a term inductively by:

$$y[t/x] := \begin{cases} t & \text{if } y = x \\ y & \text{if } y \neq x \end{cases}$$
$$(uv)[t/x] := (u[t/x])(v[t/x])$$
$$(\lambda z.u)[t/x] := \lambda z.\,(u[t/x]) \qquad (*)$$

where $(*)$ indicates the condition that $z \notin \mathsf{fv}(x\,t)$. ▲

Note that, due to identification of α-equivalent terms, it is always possible to rename bound variables so that condition $(*)$ is satisfied: for example,

$$(\lambda z.zx)[z/x] =_\alpha (\lambda y.yx)[z/x] = \lambda y.yz$$

We proceed to the definition of β-reduction and β-conversion, which are both relations defined on pairs of terms and express the computational content of the calculus.

Definition 31 We take ***β-reduction***, $\longrightarrow_\beta$, to be the relation induced by applying the following reduction rule inside terms.

$$(\lambda x.t)\,u \longrightarrow_\beta t[u/x]\,.$$

We take ***β-conversion***, $=_\beta$, to be the symmetric reflexive transitive closure of β-reduction, that is the equivalence relation on terms induced by:

$$(\lambda x.t)\,u =_\beta t[u/x]\,.$$

▲

With β-reduction we obtain a notion of 'computational dynamics'. For example,

$$(\lambda f.\lambda g.\,f\,g\,x)\,y\,y \longrightarrow (\lambda g.\,y\,g\,x)\,y \longrightarrow y\,y\,x\,.$$

1.6.3 Simply-typed $\boldsymbol{\lambda}$-calculus

The 'pure' λ-calculus we have discussed so far is *very* unconstrained. For example, it allows *self-application*, *i.e.* and terms like xx are perfectly legal. On the one hand, this means that the calculus very expressive: for example, we can encode *recursion* by setting

$$\mathbf{Y} := \lambda f.\,(\lambda x.\,f(xx))(\lambda x.\,f(xx))\,.$$

We have:

$$\mathbf{Y}t \to (\lambda x.\, t(xx))(\lambda x.\, t(xx)) \to t((\lambda x.\, t(xx))(\lambda x.\, t(xx))) = t(\mathbf{Y}t)\,.$$

However, self-application also leads to *divergences*. For example, setting $\Omega := \omega\omega$,

$$\Omega \longrightarrow \Omega \longrightarrow \Omega \longrightarrow \cdots\,.$$

Historically, Curry extracted $\mathbf{Y}$ from an analysis of Russell's Paradox, so this should be no surprise.

The solution is to introduce *types*. The original idea, due to Church following Russell, was that:

Types are there to stop you doing bad things

However, it has turned out that types constitute one of the most fruitful *positive* ideas in Computer Science, and provide one of the key disciplines of programming.

Definition 32 Let us assume a set of *base types*, ranged over by b. The ***simply-typed*** λ***-calculus*** is defined as follows.

Type	$\mathrm{TY} \ni T, U \ ::= \ b \mid T \to U \mid T \times U$
Term	$\mathrm{TE} \ni t, u \ ::= \ x \mid t\,u \mid \lambda x.\, t \mid \langle t, u\rangle \mid \pi_1 u \mid \pi_2 u$
Typing context	$\Gamma \ ::= \ \varnothing \mid x : T, \Gamma$

Note that $x : T, \Gamma$ stands for $\{x : T\} \cup \Gamma$ with x not appearing in Γ.
Examples of types:

$\iota \to \iota \to \iota$	first-order function type
$(\iota \to \iota) \to \iota$	second-order function type

Terms are typed by deriving typing judgements of the form $\Gamma \vdash t : T$, which is to be understood as the assertion that term t has the type T under the assumptions that x_1 has type T_1, ..., t_k has type T_k, where $\Gamma = x_1 : T_1, \ldots, x_k : T_k$.

The System of Simply-Typed λ*-calculus*

The following set of rules are used to derive typing judgements.

Variable	Product	Function
$\dfrac{}{\Gamma, x : T \vdash x : T}$	$\dfrac{\Gamma \vdash t : T \qquad \Gamma \vdash u : U}{\Gamma \vdash \langle t, u\rangle : T \times U}$ $\dfrac{\Gamma \vdash v : T \times U}{\Gamma \vdash \pi_1 v : T}$ $\dfrac{\Gamma \vdash v : T \times U}{\Gamma \vdash \pi_2 v : U}$	$\dfrac{\Gamma, x : U \vdash t : T}{\Gamma \vdash \lambda x.\, t : U \to T}$ $\dfrac{\Gamma \vdash t : U \to T \qquad \Gamma \vdash u : U}{\Gamma \vdash tu : T}$

▲

We say that a term t ***is typable*** iff there is a context Γ and a type T such that $\Gamma \vdash t : T$ is *derivable*.

Exercise 39. Can you type the following terms?

$$\lambda x.\, xx\,, \qquad \lambda f.\, (\lambda x.\, f(xx))(\lambda x.\, f(xx))\,.$$

The reduction and conversion rules in the calculus now involve also rules for products. We also include η-rules, which are essentially extensionality principles.

Definition 33 We define β-reduction, $\longrightarrow_\beta$, by the following rules, and let β-conversion, $=_\beta$, be its symmetric reflexive transitive closure.

$$\begin{aligned} (\lambda x.\, t)u &\longrightarrow_\beta t[u/x] \\ \pi_1\langle t, u\rangle &\longrightarrow_\beta t \\ \pi_2\langle t, u\rangle &\longrightarrow_\beta u \end{aligned}$$

Moreover, η-conversion, $=_\eta$, is the congruence equivalence relation defined by the following rules,

$$\begin{aligned} t &=_\eta \lambda x.\, tx && x \notin \mathsf{fv}(t), \text{ at function types} \\ v &=_\eta \langle \pi_1 v, \pi_2 v\rangle && \text{at product types} \end{aligned}$$

and λ-conversion, $=_\lambda$, is the transitive closure of $=_\beta \cup =_\eta$. ▲

Exercise 40 (Weakening). Let t be a λ-term, Γ a context and x some variable not appearing in Γ. Show that, for any type U, if $\Gamma \vdash t : T$ is typed then so is $\Gamma, x : U \vdash t : T$.

Exercise 41 (Subject Reduction). Show first that Cut is admissible in the typing system of the simply-typed λ-calculus:

$$\frac{\Gamma \vdash t : T \qquad \Gamma, x : T \vdash u : U}{\Gamma \vdash u[t/x] : U}\ \mathsf{Cut}$$

Show then that, for any typed term $\Gamma \vdash t : T$, if $t \to_\beta t'$ then $\Gamma \vdash t' : T$ is typed.

The Curry-Howard correspondence

Comparing the following two systems,

- Natural Deduction System for $\wedge$, $\supset$,
- Simple Type System for $\times$, $\rightarrow$,

we notice that if we equate

$$\wedge \equiv \times$$
$$\supset \equiv \rightarrow$$

then they are the same! This is the *Curry-Howard isomorphism* (sometimes: 'Curry-Howard correspondence'). It works on three levels:

Formulas	Types
Proofs	Terms
Proof transformations	Term reductions

Strong Normalisation

Term reduction results in a *normal form*: an *explicit but much longer expression* which corresponds to a proof in which all lemmas have been eliminated. Even simply typed lambda calculus has enormous (*non-elementary*) complexity.

A λ-term is called a ***redex*** if it is in one of forms of the left-hand-side of the β-reduction rules and therefore β-reduction can be applied to it. A term is in ***normal form*** if it contains no redexes as subterms.

Fact 34 (SN) *For every term t, there is no infinite sequence of β-reductions:*

$$t \longrightarrow t_0 \longrightarrow t_1 \longrightarrow t_2 \longrightarrow \cdots$$

Constructive reading of formulas

The view of proofs as containing computational content can be also detected in the *Brouwer-Heyting-Kolmogorov interpretation* of intuitionistic logic:

- A proof of an implication $A \supset B$ is a procedure which transforms any proof of A into a proof of B.
- A proof of $A \wedge B$ is a pair consisting of a proof of A and a proof of B.

These readings motivate identifying $A \wedge B$ with $A \times B$, and $A \supset B$ with $A \rightarrow B$.

Moreover, these ideas have strong connections to computing. The λ-calculus is a 'pure' version of functional programming languages such as Haskell and SML. So we get a reading of:

Proofs as Programs

We now have our link between Logic and Computation. We proceed to complete the triangle which opened this section by showing their connection to Categories.

1.6.4 Categories

We establish the link from Logic (and Computation) to Categories. Let $\mathcal{C}$ be a cartesian closed category. We shall interpret formulas (or types) as objects of $\mathcal{C}$. A morphism $f : A \to B$ will then correspond to a proof of B from assumption A, i.e. a proof of $A \vdash B$ (a typed term $x : A \vdash t : B$).

Note that the bare structure of a category only supports proofs from a single assumption. Since $\mathcal{C}$ has finite products, a proof of

$$A_1, \ldots, A_k \vdash A$$

will correspond to a morphism

$$f : A_1 \times \cdots \times A_k \longrightarrow A\,.$$

The correspondence is depicted as follows.

Axiom	$\dfrac{}{\Gamma, A \vdash A}\ \mathsf{Id}$	$\dfrac{}{\pi_2 : \Gamma \times A \longrightarrow A}$
Conjunction	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge\mathsf{I}$	$\dfrac{f : \Gamma \longrightarrow A \qquad g : \Gamma \longrightarrow B}{\langle f, g \rangle : \Gamma \longrightarrow A \times B}$
	$\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge\mathsf{E}_1$	$\dfrac{f : \Gamma \longrightarrow A \times B}{\pi_1 \circ f : \Gamma \longrightarrow A}$
	$\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge\mathsf{E}_2$	$\dfrac{f : \Gamma \longrightarrow A \times B}{\pi_2 \circ f : \Gamma \longrightarrow B}$
Implication	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ \supset\mathsf{I}$	$\dfrac{f : \Gamma \times A \longrightarrow B}{\Lambda(f) : \Gamma \longrightarrow (A \Rightarrow B)}$
	$\dfrac{\Gamma \vdash A \supset B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \supset\mathsf{E}$	$\dfrac{f : \Gamma \longrightarrow (A \Rightarrow B) \quad g : \Gamma \to A}{\mathsf{ev}_{A,B} \circ \langle f, g \rangle : \Gamma \longrightarrow B}$

Moreover, the rules for β- and η-conversion are all then *derivable* from the equations of cartesian closed categories. So cartesian closed categories are *models* of $\wedge$,$\supset$-logic at the level of proofs and proof-transformations, and of simply typed λ-calculus at the level of terms and term-conversions. The connection to computation is examined in more detail below.

1.6.5 Categorical semantics of simply-typed λ-calculus

We translate the simply-typed λ-calculus into a cartesian closed category $\mathcal{C}$, so that to each typed term $x_1 : T_1, ..., x_k : T_k \vdash t : T$ corresponds an arrow

$$[\![t]\!] : [\![T_1]\!] \times \cdots \times [\![T_k]\!] \longrightarrow [\![T]\!]\,.$$

The translation if given by the function $[\![\_]\!]$ defined below ('semantic brackets').

Definition 35 (Semantic translation) Let $\mathcal{C}$ be a CCC and suppose we are given an assignment of an object $\tilde{b}$ to each base type b. Then, the translation is defined recursively on types by:

$$[\![b]\!] := \tilde{b}\,, \quad [\![T \times U]\!] := [\![T]\!] \times [\![U]\!]\,, \quad [\![T \to U]\!] := [\![T]\!] \Rightarrow [\![U]\!]\,.$$

and on typed terms by:

$$\frac{}{[\![\Gamma, x : T \vdash x : T]\!] := \pi_2 : [\![\Gamma]\!] \times [\![T]\!] \longrightarrow [\![T]\!]}$$

$$\frac{[\![\Gamma \vdash t : T \times U]\!] = f : [\![\Gamma]\!] \longrightarrow [\![T]\!] \times [\![U]\!]}{[\![\Gamma \vdash \pi_1 t : T]\!] := [\![\Gamma]\!] \xrightarrow{f} [\![T]\!] \times [\![U]\!] \xrightarrow{\pi_1} [\![T]\!]}$$

$$\frac{[\![\Gamma \vdash t : T]\!] = f : [\![\Gamma]\!] \longrightarrow [\![T]\!] \qquad [\![\Gamma \vdash u : U]\!] = g : [\![\Gamma]\!] \longrightarrow [\![U]\!]}{[\![\Gamma \vdash \langle t, u\rangle : T \times U]\!] := [\![\Gamma]\!] \xrightarrow{\langle f,g\rangle} [\![T]\!] \times [\![U]\!]}$$

$$\frac{[\![\Gamma, x : T \vdash t : U]\!] = f : [\![\Gamma]\!] \times [\![T]\!] \longrightarrow [\![U]\!]}{[\![\Gamma \vdash \lambda x.\, t : T \to U]\!] := \Lambda(f) : [\![\Gamma]\!] \longrightarrow ([\![T]\!] \Rightarrow [\![U]\!])}$$

$$\frac{[\![\Gamma \vdash t : T \to U]\!] = f \qquad [\![\Gamma \vdash u : T]\!] = g}{[\![\Gamma \vdash t\, u : U]\!] := [\![\Gamma]\!] \xrightarrow{\langle f,g\rangle} ([\![T]\!] \Rightarrow [\![U]\!]) \times [\![T]\!] \xrightarrow{\mathsf{ev}} [\![U]\!]}$$

▲

Our aim now is to verify that the λ-conversion (induced by β- and η-rules) is preserved by the translation, i.e. that, for any t, u,

$$t =_\lambda u \implies [\![t]\!] = [\![u]\!]\,.$$

Let us recall some structures from CCC's. Given $f_1 : D_1 \to E_1$, $f_2 : D_2 \to E_2$, we defined

$$f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2\rangle : D_1 \times D_2 \longrightarrow E_1 \times E_2\,,$$

and we showed that $(f_1 \times f_2) \circ \langle h_1, h_2\rangle = \langle f_1 \circ h_1, f_2 \circ h_2\rangle$. Moreover, exponentials are given by the following natural bijection,

$$\frac{f : D \times E \longrightarrow F}{\Lambda(f) : D \longrightarrow (E \Rightarrow F)}$$

and recall the basic equation:

$$\mathsf{ev} \circ \langle \Lambda(f) \times \mathsf{id}_E\rangle = f\,.$$

Moreover, $\Lambda(f)$ is the unique function $D \to (E \Rightarrow F)$ satisfying this equation, with uniqueness being specified by the equation:

$$\forall h : D \longrightarrow (E \Rightarrow F).\ \Lambda(\mathsf{ev} \circ (h \times \mathsf{id}_E)) = h\,.$$

Proposition 8 (A key property of currying). *For any* $f : A \times B \to C$ *and* $g : A' \to A$,

$$\Lambda(f) \circ g = \Lambda(f \circ (g \times \mathsf{id}_B)) .$$

Proof:

$$\begin{aligned}\Lambda(f) \circ g &= \Lambda(\mathsf{ev} \circ ((\Lambda(f) \circ g) \times \mathsf{id}_B)) \\ &= \Lambda(\mathsf{ev} \circ (\Lambda(f) \times \mathsf{id}_B) \circ (g \times \mathsf{id}_B))) \\ &= \Lambda(f \circ (g \times \mathsf{id}_B)) .\end{aligned}$$

■

Substitution Lemma

We consider a *simultaneous substitution* for all the free variables in a term.

Definition 36 Let $\Gamma := \{x_1 : T_1, \ldots, x_k : T_k\}$. Given typed terms

$$\Gamma \vdash t : T \quad \text{and} \quad \Gamma \vdash t_i : T_i\,,\ 1 \le i \le k\,,$$

we define $t[\mathbf{t}/\mathbf{x}] \equiv t[t_1/x_1, \ldots, t_k/x_k]$ recursively by:

$$\begin{aligned} x_i[\mathbf{t}/\mathbf{x}] &:= t_i \\ (\pi_i\, t)[\mathbf{t}/\mathbf{x}] &:= \pi_i(t[\mathbf{t}/\mathbf{x}]) \\ \langle t, u\rangle[\mathbf{t}/\mathbf{x}] &:= \langle t[\mathbf{t}/\mathbf{x}], u[\mathbf{t}/\mathbf{x}]\rangle \\ (t\, u)[\mathbf{t}/\mathbf{x}] &:= (t[\mathbf{t}/\mathbf{x}])(u[\mathbf{t}/\mathbf{x}]) \\ (\lambda x.\, t)[\mathbf{t}/\mathbf{x}] &:= \lambda x.\, t[\mathbf{t}, x/\mathbf{x}, x] . \end{aligned}$$

▲

Note that, in contrast to ordinary substitution, simultaneous substitution can be defined directly on *raw terms*, that is prior to equating them modulo α-equivalence. Moreover, we can show that:

$$t[t_1/x_1, \ldots, t_k/x_k] = t[t_1/x_1] \cdots [t_k/x_k] .$$

We can now show the following *Substitution Lemma*.

Proposition 9. *For* $t, t_1, \ldots, t_k$ *as in the previous definition,*

$$[\![t[t_1/x_1, \ldots, t_k/x_k]]\!] = [\![t]\!] \circ \langle [\![t_1]\!], \ldots, [\![t_k]\!]\rangle .$$

Proof: By induction on the structure of t.
(1) If $t = x_i$:

$$[\![x_i[\mathbf{t}/\mathbf{x}]]\!] = [\![t_i]\!] = \pi_i \circ \langle [\![t_1]\!], \ldots, [\![t_k]\!]\rangle = [\![x_i]\!] \circ \langle [\![t_1]\!], \ldots, [\![t_k]\!]\rangle .$$

(2) If $t = uv$ then, abbreviating $\langle [\![t_1]\!], \ldots, [\![t_k]\!]\rangle$ to $\langle [\![\mathbf{t}]\!]\rangle$ we have:

$$\begin{aligned} [\![uv[\mathbf{t}/\mathbf{x}]]\!] &= [\![(u[\mathbf{t}/\mathbf{x}])(v[\mathbf{t}/\mathbf{x}])]\!] && \text{Defn of substitution} \\ &= \mathsf{ev} \circ \langle [\![u[\mathbf{t}/\mathbf{x}]]\!], [\![v[\mathbf{t}/\mathbf{x}]]\!]\rangle && \text{Defn of semantic function} \\ &= \mathsf{ev} \circ \langle [\![u]\!] \circ [\![\mathbf{t}]\!], [\![v]\!] \circ [\![\mathbf{t}]\!]\rangle && \text{Induction hyp.} \\ &= \mathsf{ev} \circ \langle [\![u]\!], [\![v]\!]\rangle \circ \langle [\![\mathbf{t}]\!]\rangle && \text{property of products} \\ &= [\![uv]\!] \circ \langle [\![\mathbf{t}]\!]\rangle && \text{Defn of semantic function} \end{aligned}$$

(3) If $t = \lambda x.\, u$:

$$
\begin{aligned}
[\![\lambda x.u[\mathbf{t}/\mathbf{x}]]\!] &= [\![\lambda x.(u[\mathbf{t}, x/\mathbf{x}, x])]\!] && \text{Defn. of substitution}\\
&= \Lambda([\![u[\mathbf{t}, x/\mathbf{x}, x]]\!]) && \text{Defn. of semantic function}\\
&= \Lambda([\![u]\!] \circ ([\![\mathbf{t}]\!] \times \mathsf{id})) && \text{Induction hyp.}\\
&= \Lambda([\![u]\!]) \circ \langle [\![\mathbf{t}]\!] \rangle && \text{Prop. 8}\\
&= [\![\lambda x.u]\!] \circ \langle [\![\mathbf{t}]\!] \rangle && \text{Defn. of semantic function}
\end{aligned}
$$

(4,5) The cases of projections and pairs are left as exercise. ■

Validating the conversion rules

We can now show that the conversion rules of the λ-calculus are *preserved* by the translation, and hence the interpretation is *sound*.

- For β-conversion: $\left[(\lambda x.\, t)u = t[u/x]\, ,\ \pi_1\langle t, u\rangle = t\, ,\ \pi_2\langle t, u\rangle = u\right]$

$$
\begin{aligned}
[\![(\lambda x.\, t)u]\!] &= \mathsf{ev} \circ \langle \Lambda([\![t]\!]), [\![u]\!] \rangle && \text{Defn. of semantics}\\
&= \mathsf{ev} \circ (\Lambda([\![t]\!]) \times \mathsf{id}) \circ \langle \mathsf{id}_{[\![\Gamma]\!]}, [\![u]\!] \rangle && \text{Property of } \times\\
&= [\![t]\!] \circ \langle \mathsf{id}_{[\![\Gamma]\!]}, [\![u]\!] \rangle && \text{Defn. of } \Lambda\\
&= [\![t[\mathbf{x}, u/\mathbf{x}, x]]\!] && \text{Substitution lemma.}
\end{aligned}
$$

$$[\![\pi_1\langle t, u\rangle]\!] = \pi_1 \circ [\![\langle t, u\rangle]\!] = \pi_1 \circ \langle [\![t]\!], [\![u]\!] \rangle = [\![t]\!]\, .$$

- For η-conversion: $\left[t = \lambda x.\, tx\, ,\ \langle \pi_1 t, \pi_2 t\rangle = t\right]$

$$[\![\lambda x.\, tx]\!] = \Lambda(\mathsf{ev} \circ ([\![t]\!] \times \mathsf{id})) = [\![t]\!] \qquad \text{Uniqueness equation } (\Rightarrow)$$

$$[\![\langle \pi_1 t, \pi_2 t\rangle]\!] = \langle \pi_1 \circ [\![t]\!], \pi_2 \circ [\![t]\!] \rangle = [\![t]\!] \qquad \text{Uniqueness equation } (\times)$$

1.6.6 Completeness?

Certainly, in a general CCC $\mathcal{C}$ there may be equalities which are *not* reflected by the semantic translation:

$$[\![t]\!] = [\![u]\!] \quad \text{yet} \quad t \neq_\lambda u\, .$$

We will now *construct* a CCC $\mathcal{C}_\lambda$ in which all equalities between arrows are translations of λ-conversions between terms.

Definition 37 We define an equivalence relations of *typed* terms by setting $(x, t) \sim_{T,U} (y, u)$ iff $x : T \vdash t : U$ and $y : T \vdash u : U$ are derivable and

$$t =_\lambda u[x/y]\, .$$

This yields an equivalence relation, so we set:

$$[(x, t)]_{T,U} := \{\ (y, u) \mid (x, t) \sim_{T,U} (y, u)\ \}$$

Similarly, $(\,\centerdot\,, t) \sim_{\centerdot,U} (\,\centerdot\,, u)$ iff $\vdash t : U$ and $\vdash u : U$ are derivable and $t =_\lambda u[x/y]$. Moreover,

$$[(\,\centerdot\,, t)]_{\centerdot,U} := \{\ (\,\centerdot\,, u) \mid (\,\centerdot\,, t) \sim_{\centerdot,U} (\,\centerdot\,, u)\ \}\,.$$

▲

We denote $[(x, t)]_{T,U}$ simply as $[x, t]$; and $[(\,\centerdot\,, t)]_{\centerdot,U}$ simply as $[\,\centerdot\,, t]$. (*Note these have nothing to do with copairings!*) We proceed with $\mathcal{C}_\lambda$.

Definition 38 $\mathcal{C}_\lambda$ is defined as follows.

Objects	$Ob(\mathcal{C}_\lambda) := \{\mathbf{1}\} \cup \{\ \widetilde{T} \mid T \text{ a } \lambda\text{-type}\ \}$
Arrows	$\mathcal{C}_\lambda(\widetilde{T}, \widetilde{U}) := \{\ [x, t] \mid x : T \vdash t : U \ \text{ is derivable}\ \}$ $\mathcal{C}_\lambda(\mathbf{1}, \widetilde{U}) := \{\ [\,\centerdot\,, t] \mid\ \vdash t : U \ \text{ is derivable}\ \}$ $\mathcal{C}_\lambda(A, \mathbf{1}) := \{\ \tau_A\ \}$
Identities	$\mathsf{id}_{\widetilde{T}} := [x, x]\,, \quad \mathsf{id}_{\mathbf{1}} := \tau_{\mathbf{1}}$
Composition	$[x, t] \circ [y, u] := [y, t[u/x]] \qquad (y \neq x)$ $[x, t] \circ [\,\centerdot\,, u] := [\,\centerdot\,, t[u/x]]$ $[\,\centerdot\,, t] \circ \tau_A := \begin{cases} [y, t] & \text{if } A = \widetilde{U} \\ [\,\centerdot\,, t] & \text{if } A = \mathbf{1} \end{cases}$ $\tau_B \circ h := \tau_A \qquad (h \in \mathcal{C}_\lambda(A, B))$

▲

Proposition 10. $\mathcal{C}_\lambda$ *is a category.*

Proof: It is not difficult to see that id's are identities. For associativity, we show the most interesting case (and leave the rest as an exercise):

$$[x, t] \circ ([y, u] \circ [z, v]) = [x, t] \circ [z, u[v/y]] = [z, t[(u[v/y])/x]]\,,$$

$$([x, t] \circ [y, u]) \circ [z, v] = [y, t[u/x]] \circ [z, v] = [z, t[u/x][v/y]]\,.$$

Since $y \neq x$ and t has at most x as a free variable, y is not free in t and therefore:

$$t[u/x][v/y] = t[(u[v/y])/x]$$

■

.

Proposition 11. $\mathcal{C}_\lambda$ *has finite products.*

Proof: Clearly, $\mathbf{1}$ is terminal with canonical arrows $\tau_A : A \longrightarrow \mathbf{1}$. For (binary) products, $\mathbf{1} \times A = A \times \mathbf{1} = A$. Otherwise, define $\widetilde{T} \xleftarrow{\pi_1} \widetilde{T} \times \widetilde{U} \xrightarrow{\pi_2} \widetilde{U}$ by:

$$\begin{aligned} \widetilde{T} \times \widetilde{U} &:= \widetilde{T \times U} \\ \pi_i &:= [x, \pi_i x] \quad i = 1, 2\,. \end{aligned}$$

Given $\widetilde{T} \xleftarrow{[x,t]} \widetilde{V} \xrightarrow{[x,u]} \widetilde{U}$, take $\langle [x,t], [x,u] \rangle : \widetilde{V} \longrightarrow \widetilde{T} \times \widetilde{U} := [x, \langle t, u \rangle]$. Then:

$$\begin{aligned} \pi_1 \circ \langle [x,t],[x,u] \rangle &= [y, \pi_1 y] \circ [x, \langle t,u \rangle] && \text{Definitions} \\ &= [x, \pi_1 \langle t,u \rangle] && \text{Defn of composition} \\ &= [x,t] && \beta\text{-conversion} \end{aligned}$$

Uniqueness left as exercise. ■

Proposition 12. *$\mathcal{C}_\lambda$ has exponentials.*

Proof: We have that $\mathbf{1} \Rightarrow A = A$ and $A \Rightarrow \mathbf{1} = \mathbf{1}$, with obvious evaluation arrows. Otherwise,

$$\begin{aligned} \widetilde{U} \Rightarrow \widetilde{T} &:= \widetilde{U \to T} \\ \mathsf{ev}_{\widetilde{U},\widetilde{T}} : (\widetilde{U} \Rightarrow \widetilde{T}) \times \widetilde{U} &\longrightarrow \widetilde{T} := [x, (\pi_1 x)(\pi_2 x)] \end{aligned}$$

Given $[x,t] : \widetilde{V} \times \widetilde{U} \longrightarrow \widetilde{T}$, take $\Lambda([x,t]) := [x_1, \lambda x_2.t[\langle x_1, x_2 \rangle / x]]$.
Then,

$$\begin{aligned} \mathsf{ev} \circ \Lambda([x,t]) \times \mathsf{id} &= \mathsf{ev} \circ \langle \Lambda([x,t]) \circ \pi_1, \mathsf{id} \circ \pi_2 \rangle \\ &= \mathsf{ev} \circ \langle [x_1, \lambda x_2.t[\langle x_1, x_2 \rangle/x]] \circ [y, \pi_1 y], [y, \pi_2 y] \rangle \\ &= \mathsf{ev} \circ \langle [y, \lambda x_2.t[\langle \pi_1 y, x_2 \rangle/x]], [y, \pi_2 y] \rangle \\ &= [z, (\pi_1 z)(\pi_2 z)] \circ [y, \underbrace{\langle \lambda x_2.t[\langle \pi_1 y, x_2 \rangle/x], \pi_2 y \rangle}_{u}] \\ &= [y, (\pi_1 u)(\pi_2 u)] \overset{\beta}{=} [y, (\lambda x_2.t[\langle \pi_1 y, x_2 \rangle/x])(\pi_2 y)] \\ &\overset{\beta}{=} [y, t[\langle \pi_1 y, \pi_2 y \rangle/x]] \overset{\eta}{=} [y, t[y/x]] = [x,t]\,. \end{aligned}$$

Uniqueness and other cases left as exercise. ■

Now applying our translation from the λ-calculus to a CCC we (can show that we) have

$$[\![\Gamma \vdash t : T]\!] = [x,\, t[\pi_i x / x_i]_{i=1..n}]$$

where $\Gamma = \{x_1 : T_1, ..., x_n : T_n\}$, $x \notin \Gamma$ and $x : \prod_{i=1..n} T_i$. Then,

$$t =_\lambda u \iff [\![\Gamma \vdash t : T]\!] = [\![\Gamma \vdash u : T]\!]\,.$$

1.6.7 Exercises

1. Give Natural Deduction proofs of the following sequents.
- $\vdash (A \supset B) \supset ((B \supset C) \supset (A \supset C))$
- $\vdash (A \supset (A \supset B)) \supset (A \supset B)$
- $\vdash (C \supset A) \supset ((C \supset B) \supset (C \supset (A \wedge B)))$
- $\vdash (A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

In each case, give the corresponding λ-term.

2. For each of the following λ-terms, find a type for it. Try to find the 'most general' type, built from 'type variables' α, β etc. For example, the most general type for the identity $\lambda x.\, x$ is $\alpha \to \alpha$. In each case, give the derivation of the type for this term (where you may assume that types can be built up from type variables as well as base types).

- $\lambda f.\, \lambda x.\, fx$
- $\lambda x.\, \lambda y.\, \lambda z.\, x(yz)$
- $\lambda x.\, \lambda y.\, \lambda z.\, xzy$
- $\lambda x.\, \lambda y.\, xyy$
- $\lambda x.\, \lambda y.\, x$
- $\lambda x.\, \lambda y.\, \lambda z.\, xz(yz)$

Reflect a little on the methods you used to do this exercise. Could they be made algorithmic?

1.7 Linearity

In the system of Natural Deduction, implicit in our treatment of assumptions in sequents

$$A_1, \ldots, A_n \vdash A$$

is that we can use them as many times as we want (including not at all). In this section we will explore the field that is opened once we apply restrictions on this approach, and thus render our treatment of assumptions more *linear* (or *resource sensitive*).

1.7.1 Gentzen sequent calculus

In order to make the manipulation of assumptions more visible, we now represent the assumptions as a *list* (possibly with repetitions) rather than a set, and use explicit structural rules to control copying, deletion and interchange of assumptions.

Definition 39 The ***structural rules*** for Logic are the following.

$$\frac{}{A \vdash A}\ \mathsf{Id}$$	$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\ \mathsf{Exch}$$
$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ \mathsf{Contr}$$	$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\ \mathsf{Weak}$$

▲

If we think of using proof rules *backwards* to reduce the task of proving a given sequent to various sub-tasks, then we see that the Contraction rule lets us *duplicate* premises, and the Weakening rule lets us *discard* them, while the Exchange rule merely lets us re-order them. The Identity axiom as given here is equivalent to the one with auxiliary premises given previously in the presence of Weakening. The structural rules have clear categorical meanings in a category $\mathcal{C}$ with products. Recalling the *diagonal transformation* $\Delta_A := \langle \mathsf{id}_A, \mathsf{id}_A \rangle$ and the *symmetry transformation* $s_{A,B} := \langle \pi_2, \pi_1 \rangle$, the meanings are as follows.

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\ \mathsf{Exch} \qquad \frac{f : \Gamma \times A \times B \times \Delta \longrightarrow C}{f \circ (\mathsf{id}_\Gamma \times s_{A,B} \times \mathsf{id}_\Delta) : \Gamma \times B \times A \times \Delta \longrightarrow C}$$

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ \mathsf{Contr} \qquad \frac{f : \Gamma \times A \times A \longrightarrow B}{f \circ (\mathsf{id}_\Gamma \times \Delta_A) : \Gamma \times A \longrightarrow B}$$

$$\frac{\Gamma \vdash B}{\Gamma, A \vdash B}\ \mathsf{Weak} \qquad \frac{f : \Gamma \longrightarrow B}{f \circ \pi_1 : \Gamma \times A \longrightarrow B}$$

Definition 40 We define the ***Gentzen sequent calculus*** for $\wedge, \supset$ as the proof system obtained by the structural rules (def. 37) and the following rules for connectives.

Conjunction	**Implication**	**Cut**
$$\frac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B}\ \wedge\mathsf{R}$$	$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ \supset\mathsf{R}$$	$$\frac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\ \mathsf{Cut}$$
$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C}\ \wedge\mathsf{L}$$	$$\frac{\Gamma \vdash A \qquad B, \Delta \vdash C}{\Gamma, A \supset B, \Delta \vdash C}\ \supset\mathsf{L}$$	

▲

Note that the sequent calculus introduces a new kind of pattern for proof rules: **Left** and **Right** rules, rather than the Introduction and Elimination rules of Natural deduction.

For example, the proof of $\supset$-transitivity is now given as follows.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Id} \qquad \dfrac{}{B \vdash B}\,\mathsf{Id}}{A, A \supset B \vdash B}\,{\supset}\mathsf{R}}{A \supset B, A \vdash B}\,\mathsf{Exch} \qquad \dfrac{}{C \vdash C}\,\mathsf{Id}}{A \supset B, A, B \supset C \vdash C}\,{\supset}\mathsf{L}}{\dfrac{A \supset B, B \supset C, A \vdash C}{A \supset B, B \supset C \vdash A \supset C}\,{\supset}\mathsf{R}}\,\mathsf{Exch}
$$

The Cut rule allows the use of *lemmas* in proofs. It also yields a *dynamics of proofs* via *Cut Elimination*, that is a dynamics of proof transformations towards the goal of eliminating the uses of the Cut rule in a proof, *i.e.* removing all lemmas and making the proof completely "explicit", meaning Cut-free. Such transformations are always possible, which leads to the following seminal result of Gentzen (*Hauptsatz*).

Fact 41 (Cut Elimination) *The* Cut *rule is admissible in the Gentzen sequent calculus without* Cut.

Exercise 42. Show that the Gentzen-rules are admissible in Natural Deduction. Moreover, show that the Natural Deduction rules are admissible in the Gentzen sequent calculus.

1.7.2 Linear Logic

In the presence of the structural rules, the Gentzen sequent calculus is entirely equivalent to the Natural Deduction system we studied earlier. Nevertheless,

> What happens if we **drop** the Contraction and Weakening rules (but keep the Exchange rule)?

It turns out we can still make good sense of the resulting proofs, terms and categories, but now in the setting of a different, 'resource-sensitive' logic.

Definition 42 ***Multiplicative Linear logic*** is a variant of standard logic with *linear* logical connectives (and linear proof rules). The multiplicative connectives for conjunction and implication are $\otimes$ and $\multimap$. Proof sequents are of the form $\Gamma \vdash A$, where Γ is now a *multiset*. The proof rules for $\otimes$,$\multimap$-Linear Logic are the multiplicative versions of the Gentzen rules.

Conjunction	**Implication**	**Cut**
$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,\otimes\mathsf{R}$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,\multimap\mathsf{R}$	$\dfrac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\,\mathsf{Cut}$
$\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\,\otimes\mathsf{L}$	$\dfrac{\Gamma \vdash A \qquad B, \Delta \vdash C}{\Gamma, A \multimap B, \Delta \vdash C}\,\multimap\mathsf{L}$	

▲

Multiplicativity here means the use of *disjoint* (i.e. non-overlapping) contexts. The use of multisets allows us to omit explicit use of the Exchange rule in our proof system. Note the following:

- The use of *disjoint* (i.e. non-overlapping) contexts.
- The given system satisfies Cut-elimination, and this leans heavily on the $\multimap\mathsf{L}$ rule. We could have used instead the following rule,

$$\frac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ \multimap\mathsf{E}$$

  which is more intuitive computationally, but then cut-elimination would fail. Note, though, that:

$$\multimap\mathsf{L}\,,\ \mathsf{Cut}\,,\ \mathsf{Id}\ \equiv\ \multimap\mathsf{E}\,,\ \mathsf{Cut}\,,\ \mathsf{Id}\,.$$

  This is shown as follows.

$$\dfrac{\dfrac{\Gamma \vdash A \quad \dfrac{}{A \multimap B \vdash A \multimap B}\,\mathsf{Id}}{\Gamma, A \multimap B \vdash B}\,\multimap\mathsf{E} \qquad B, \Delta \vdash C}{\Gamma, A \multimap B, \Delta \vdash C}\,\mathsf{Cut} \qquad \dfrac{\Gamma \vdash A \multimap B \quad \dfrac{\dfrac{}{B \vdash B}\,\mathsf{Id} \quad \Delta \vdash A}{A \multimap B, \Delta \vdash B}\,\multimap\mathsf{L}}{\Gamma, \Delta \vdash B}\,\mathsf{Cut}$$

Exercise 43. Can you construct proofs in Linear Logic of the following sequents? (Hint: Use the Cut Elimination property.)

- $A \vdash A \otimes A$
- $(A \otimes A) \multimap B \vdash A \multimap B$
- $\vdash A \multimap (B \multimap A)$

Related to linear logic is the *linear* λ*-calculus*, which is a linear version of the simply-typed λ-calculus.

Definition 43 The ***linear*** $\boldsymbol{\lambda}$***-calculus*** is defined as follows.

Type	$\mathrm{TY} \ni T, U \ ::= \ b \mid T \multimap U \mid T \otimes U$
Term	$\mathrm{TE} \ni t, u \ ::= \ x \mid t\,u \mid \lambda x.\,t \mid t \otimes u \mid \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ t$
Typing context	$\Gamma \ ::= \ \varnothing \mid x : T, \Gamma$

Note that $x : T, \Gamma$ stands for $\{x : T\} \cup \Gamma$ with x not appearing in Γ. Terms are typed by use of *typing rules* as follows.

$\dfrac{}{x : T \vdash x : T}$	$\dfrac{\Gamma \vdash t : T \qquad x : T, \Delta \vdash u : U}{\Gamma, \Delta \vdash u[t/x] : U}$
$\dfrac{\Gamma, x : U \vdash t : T}{\Gamma \vdash \lambda x.\, t : U \multimap T}$	$\dfrac{\Gamma \vdash t : U \qquad x : U, \Delta \vdash u : V}{\Gamma, f : T \multimap U, \Delta \vdash u[ft/x] : V}$
$\dfrac{\Gamma \vdash t : T \qquad \Delta \vdash u : U}{\Gamma, \Delta \vdash t \otimes u : T \otimes U}$	$\dfrac{\Gamma, x : T, y : U \vdash v : V}{\Gamma, z : T \otimes U \vdash \mathsf{let}\ z\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ v : V}$

Note that Cut-free proofs *always yield terms in normal form.*
The rules for β-reduction are:

$$(\lambda x.\, t)u \longrightarrow_\beta t[u/x]$$
$$\mathsf{let}\ t \otimes u\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ v \longrightarrow_\beta v[t/x, u/y]\,.$$

▲

Note that term formation is now highly constrained by the form of the typing judgements. In particular,

$$x_1 : A_1, \ldots, x_k : A_k \vdash t : A$$

now implies that each x_i occurs **exactly once** (free) in t. Moreover, note that, for function application, instead of the rule on the LHS below, we could have used the more intuitive rule on the RHS.

$$\frac{\Gamma \vdash t : U \qquad x : U, \Delta \vdash u : V}{\Gamma, f : T \multimap U, \Delta \vdash u[ft/x] : V} \qquad\qquad \frac{\Gamma \vdash t : A \multimap B \qquad \Delta \vdash u : A}{\Gamma, \Delta \vdash t\, u : B}$$

As we did in the logic, we can show that the typing systems with one or the other rule are equivalent.

1.7.3 Linear Logic in monoidal categories

We proceed to give a categorical counterpart to linearity by providing a categorical interpretation of linear logic. Note that CCC's are no longer adequate for this task, as they contain arrows

$$\Delta_A : A \longrightarrow A \times A\,, \quad \pi_1 : A \times B \longrightarrow A$$

which violate linearity. It turns out that the right setting is that of *symmetric monoidal closed categories.*

Definition 44 A ***monoidal category*** is a structure $(\mathcal{C}, \otimes, I, a, l, r)$ where:

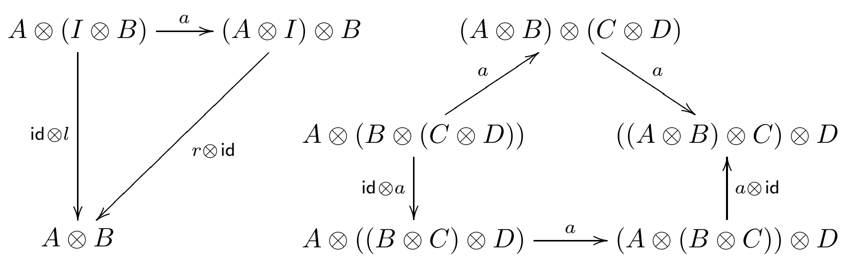
- $\mathcal{C}$ is a category,
- $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ is a functor (*tensor*),
- I is a distinguished object of $\mathcal{C}$ (*unit*),

- a, l, r are natural isomorphisms (*structural isos*) of the types:

$$a_{A,B,C} : A \otimes (B \otimes C) \xrightarrow{\cong} (A \otimes B) \otimes C$$

$$l_A : I \otimes A \xrightarrow{\cong} A \qquad r_A : A \otimes I \xrightarrow{\cong} A$$

such that $l_I = r_I : I \otimes I \to I$ and the following diagrams commute.



▲

The monoidal diagrams ensure coherence, described by the slogan:

"... 'all' diagrams involving a, l and r must commute."

Examples:

- Both products and coproducts give rise to monoidal structures — which are the common denominator between them. (But in addition, products have *diagonals* and *projections*, and coproducts have *codiagonals* and *injections*.)
- $(\mathbb{N}, \leq, +, 0)$ is a monoidal category.
- **Rel**, the category of sets and relations, with cartesian product (which is *not* the categorical product).
- $\mathbf{Vect}_k$ with the tensor product.

Let us examine the example of **Rel** in some detail. We take $\otimes$ to be the cartesian product, which is defined on relations $R : X \to X'$ and $S : Y \to Y'$ as follows.

$$\forall (x, y) \in X \times Y, (x', y') \in X' \times Y'. \; (x, y) R \otimes S (x', y') \iff x R x' \wedge y S y' \, .$$

It is not difficult to show that this is indeed a functor. Note that, in the case that R, S are *functions*, $R \otimes S$ is the same as $R \times S$ in **Set**. Moreover, we take each $a_{A,B,C}$ to be the associativity function for products (in **Set**), which is an iso in **Set** and hence also in **Rel**. Finally, we take I to be the one-element set, and l_A, r_A to be the projection functions: their relational converses are their inverses in **Rel**. The monoidal diagrams commute simply because they commute in **Set**.

Exercise 44. Verify that $(\mathbb{N}, \leq, +, 0)$ and $\mathbf{Vect}_k$ are monoidal categories.

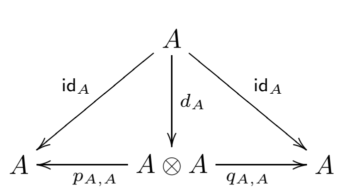
Tensors and products

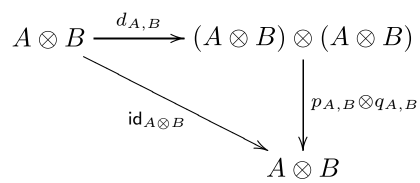
As we mentioned earlier, products are tensor with extra structure: natural diagonals and projections. This fact, which reflects *no-cloning* and *no-deleting* of Linear Logic, is shown as follows.

Proposition 13. *Let $\mathcal{C}$ be a monoidal category $(\mathcal{C}, \otimes, I, a, l, r)$. $\otimes$ induces a product structure iff there exist natural diagonals and projections, i.e. natural transformations given by arrows*

$$d_A : A \longrightarrow A \otimes A\,, \qquad p_{A,B} : A \times B \longrightarrow A\,, \qquad q_{A,B} : A \times B \longrightarrow B\,,$$

such that the following diagrams commute.





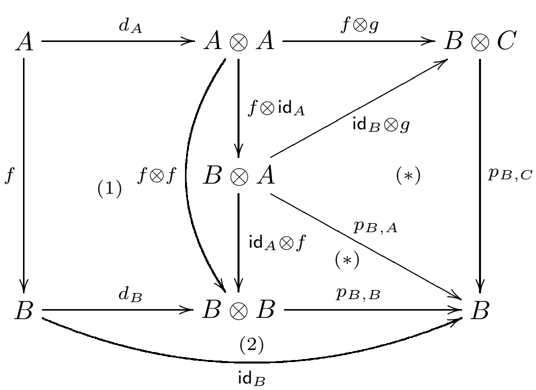
Proof: The "only if" direction is straightforward. For the converse, let $\mathcal{C}$ be monoidal with natural projections and diagonals. Then, we take product pairs to be pairs of the form

$$A \overset{p_{A,B}}{\longleftarrow} A \otimes B \overset{q_{A,B}}{\longrightarrow} B\,.$$

Moreover, for any pair of arrows $B \overset{f}{\longleftarrow} A \overset{g}{\longrightarrow} C$, define

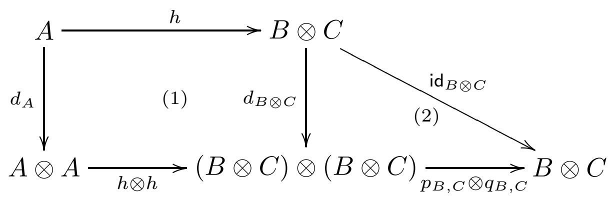
$$\langle f, g \rangle := A \overset{d_A}{\longrightarrow} A \otimes A \overset{f \otimes g}{\longrightarrow} B \otimes C\,.$$

Then the product diagram commutes. For example:



(∗) naturality of p
(1) naturality of d
(2) hypothesis

For uniqueness, if $h : A \to B \otimes C$ then the following diagram commutes,



(1) naturality of d
(2) hypothesis

so $h = \langle \pi_1 \circ h, \pi_2 \circ h \rangle$. ■

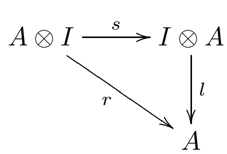
SMCC

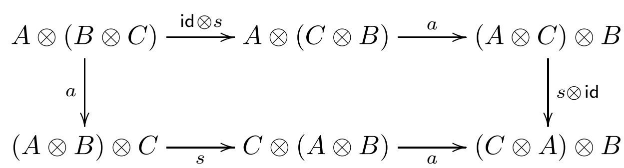
Linear Logic is interpreted in monoidal categories with two more pieces of structure: monoidal symmetries and monoidal closures. The former essentially correspond to the Exchange rule (now incorporated in the multiset specification of assumptions), while the latter realises linear implication.

Definition 45 A ***symmetric monoidal category*** is a monoidal category $(\mathcal{C}, \otimes, I, a, l, r)$ with an additional natural isomorphism (*symmetry*),

$$s_{A,B} : A \otimes B \xrightarrow{\cong} B \otimes A$$

such that $s_{B,A} = s^{-1}_{A,B}$ and the following diagrams commute.





▲

Definition 46 A ***symmetric monoidal closed category (SMCC)*** is a symmetric monoidal category $(\mathcal{C}, \otimes, I, a, l, r, s)$ such that, for each object A, there is a couniversal arrow to the functor

$$\_ \otimes A : \mathcal{C} \longrightarrow \mathcal{C}\,.$$

That is, for all pairs A, B, there is an object $A \multimap B$ and a morphism

$$\mathsf{ev}_{A,B} : (A \multimap B) \otimes A \longrightarrow B$$

such that, for every morphism $f : C \otimes A \to B$, there is a *unique* morphism $\Lambda(f) : C \to (A \multimap B)$ such that

$$\mathsf{ev}_{A,B} \circ (\Lambda(f) \otimes \mathsf{id}_A) = f\,.$$

▲

Note that, although we use notation borrowed from CCC's (ev, Λ), these are *different* structures! Examples of symmetric monoidal closed categories are **Rel**, $\mathbf{Vect}_k$, and (*a fortiori*) cartesian closed categories.

Exercise 45. Show that **Rel** is a symmetric monoidal closed category.

Linear logic in SMCC's

Just as cartesian closed categories correspond to $\wedge,\supset$-logic (and simply-typed λ-calculus), so do symmetric monoidal closed categories correspond to $\otimes,\multimap$-logic (and linear λ-calculus).

So let $\mathcal{C}$ be a symmetric monoidal closed category. The interpretation of a linear sequent

$$A_1, \ldots, A_k \vdash A$$

will be a morphism

$$f : A_1 \otimes \cdots \otimes A_k \longrightarrow A\,.$$

To be precise in our interpretation, we will again treat contexts as lists of formulas, and explicitly interpret the Exchange rule by:

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} \qquad \frac{f : \Gamma \otimes A \otimes B \otimes \Delta \longrightarrow C}{f \circ (\mathsf{id}_\Gamma \otimes s_{A,B} \otimes \mathsf{id}_\Delta) : \Gamma \otimes B \otimes A \otimes \Delta \longrightarrow C}$$

The rest of the rules are translated as follows.

$\dfrac{}{A \vdash A}$	$\dfrac{}{\mathsf{id}_A : A \longrightarrow A}$
$\dfrac{\Gamma \vdash A \qquad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}$	$\dfrac{f : \Gamma \longrightarrow A \qquad g : A \otimes \Delta \longrightarrow B}{g \circ (f \otimes \mathsf{id}_\Delta) : \Gamma \otimes \Delta \longrightarrow B}$
$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$ $\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$	$\dfrac{f : \Gamma \longrightarrow A \qquad g : \Delta \longrightarrow B}{f \otimes g : \Gamma \otimes \Delta \longrightarrow A \otimes B}$ $\dfrac{f : (\Gamma \otimes A) \otimes B \longrightarrow C}{f \circ a_{A,B,C} : \Gamma \otimes (A \otimes B) \longrightarrow C}$
$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}$ $\dfrac{\Gamma \vdash A \multimap B \qquad \Delta \vdash A}{\Gamma, \Delta \vdash B}$	$\dfrac{f : \Gamma \otimes A \longrightarrow B}{\Lambda(f) : \Gamma \longrightarrow (A \multimap B)}$ $\dfrac{f : \Gamma \longrightarrow (A \multimap B) \qquad g : \Delta \longrightarrow A}{\mathsf{ev}_{A,B} \circ (f \otimes g) : \Gamma \otimes \Delta \longrightarrow B}$

Exercise 46. Let $\mathcal{C}$ be a symmetric monoidal closed category. Give the interpretation of the $\multimap$-left rule in $\mathcal{C}$:

$$\frac{\Gamma \vdash A \qquad B, \Delta \vdash C}{\Gamma, A \multimap B, \Delta \vdash C}\ \multimap\mathsf{L}$$

1.7.4 Beyond the multiplicatives

Linear Logic has three 'levels' of connectives, each describing a different aspect of standard logic:

- The **multiplicatives**: e.g. $\otimes$, $\multimap$,
- The **additives**: *additive conjunction* & and *disjunction* $\oplus$,
- The **exponentials**, allowing controlled access to copying and discarding.

We focus on additive conjunction and the exponential "**!**", which will allow us to recover the 'expressive power' of standard $\wedge,\supset$-logic.

Definition 47 The logical connective for ***additive disjunction*** is &, and the related proof rules are the following.

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\ \&\mathsf{R} \qquad \frac{\Gamma, A \vdash C}{\Gamma, A \,\&\, B \vdash C}\ \&\mathsf{L} \qquad \frac{\Gamma, B \vdash C}{\Gamma, A \,\&\, B \vdash C}\ \&\mathsf{L}$$

▲

So additive conjunction has proof rules that are identical to those of standard conjunction ($\supset$). Note though that, since by linearity an argument of type $A \,\&\, B$ can only be used once, each use of a left rule for & makes a once-and-for-all *choice* of a projection. On the other hand, $A \otimes B$ represents a conjunction where *both* projections must be available.

Additive conjunction can be interpreted in any symmetric monoidal closed category with products, i.e. a category $\mathcal{C}$ with structure $(\otimes, \multimap, \times)$ where $(\otimes, \multimap)$ are a symmetric monoidal tensor and its adjoint, and $\times$ is a product.

$$\frac{f : \Gamma \longrightarrow A \qquad g : \Gamma \longrightarrow B}{\langle f, g\rangle : \Gamma \longrightarrow A \times B} \qquad \frac{f : \Gamma \otimes A \longrightarrow C}{f \circ (\mathsf{id} \otimes \pi_1) : \Gamma \otimes (A \times B) \longrightarrow C}$$

Moreover, we can extend the linear λ-calculus with term constructors for additive conjunction as follows.

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B}{\Gamma \vdash \langle t, u\rangle : A \,\&\, B} \quad \frac{\Gamma, x : A \vdash t : C}{\Gamma, z : A \,\&\, B \vdash \mathbf{let}\ z = \langle x, -\rangle\ \mathbf{in}\ t : C}$$

$$\frac{\Gamma, B \vdash C}{\Gamma, z : A \,\&\, B \vdash \mathbf{let}\ z = \langle -, y\rangle\ \mathbf{in}\ t : C}$$

The β-reduction rules related to these constructs are:

$$\mathbf{let}\ \langle t, u\rangle = \langle x, -\rangle\ \mathbf{in}\ v \longrightarrow_\beta v[t/x]$$
$$\mathbf{let}\ \langle t, u\rangle = \langle -, y\rangle\ \mathbf{in}\ v \longrightarrow_\beta v[u/y]\,.$$

Finally, we can gain back the lost structural rules, in *disciplined* versions, by introducing an exponential *bang* operator ! which is a kind of *modality* which enables formulas to participate in structural rules.

Definition 48 The logical connective for ***bang*** is $\mathbf{!}$, and the related proof rules are the following.

$$\frac{\Gamma, A \vdash B}{\Gamma, \mathbf{!}A \vdash B}\ \mathsf{!L} \qquad \frac{\mathbf{!}\Gamma \vdash A}{\mathbf{!}\Gamma \vdash \mathbf{!}A}\ \mathsf{!R} \qquad \frac{\Gamma \vdash B}{\Gamma, \mathbf{!}A \vdash B}\ \mathsf{Weak} \qquad \frac{\Gamma, \mathbf{!}A, \mathbf{!}A \vdash B}{\Gamma, \mathbf{!}A \vdash B}\ \mathsf{Contr}$$

▲

We can now see the discipline imposed in structural rules: in order for the rules to be applied, the participating formulas need to be tagged with bang.

Interpreting standard Natural Deduction

We are now in position to recover the standard logical connectives $\wedge$, $\supset$ within Linear Logic. If we interpret

$$\begin{aligned} A \supset B &:= \mathbf{!}A \multimap B \\ A \wedge B &:= A \,\&\, B \end{aligned}$$

and each $\wedge$,$\supset$-sequent $\Gamma \vdash A$ as $\mathbf{!}\Gamma \vdash A$, then each proof rule of Natural Deduction for $\wedge$,$\supset$ is admissible in the proof system of Linear Logic for $\otimes, \multimap, \&, \mathbf{!}$.

Note in particular that the interpretation

$$A \supset B \ := \ !A \multimap B$$

decomposes the fundamental notion of implication into finer notions — like 'splitting the atom of logic'!

1.7.5 Exercises

1. Give proofs of the following sequents in Linear Logic:
   a) $\vdash A \multimap A$
   b) $A \multimap B, B \multimap C \vdash A \multimap C$
   c) $\vdash (A \multimap B \multimap C) \multimap (B \multimap A \multimap C)$
   d) $A \otimes (B \otimes C) \vdash (A \otimes B) \otimes C$
   e) $A \otimes B \vdash B \otimes A$

   For each of the proofs constructed give:
   - the corresponding linear λ-term,
   - its interpretation in **Rel**.

2. Consider a symmetric monoidal closed category $\mathcal{C}$.
   a) Suppose the sequents $\Gamma_1 \vdash A$, $\Gamma_2 \vdash B$ and $A, B, \Delta \vdash C$ are provable and let their interpretations (i.e. the interpretations of their proofs) in $\mathcal{C}$ be $f_1 : \Gamma_1 \to A$, $f_2 : \Gamma_2 \to B$ and $g : A \otimes B \otimes \Delta \to C$ respectively. Find then the interpretations h_1, h_2 of the following proofs:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma_1 \vdash A} \quad \dfrac{\vdots}{\Gamma_2 \vdash B}}{\Gamma_1, \Gamma_2 \vdash A \otimes B}\ \otimes\mathsf{R} \qquad \dfrac{\dfrac{\vdots}{A, B, \Delta \vdash C}}{A \otimes B, \Delta \vdash C}\ \otimes\mathsf{L}}{\Gamma_1, \Gamma_2, \Delta \vdash C}\ \mathsf{Cut} \qquad \dfrac{\dfrac{\vdots}{\Gamma_2 \vdash B} \qquad \dfrac{\dfrac{\vdots}{\Gamma_1 \vdash A} \quad \dfrac{\vdots}{A, B, \Delta \vdash C}}{\Gamma_1, B, \Delta \vdash C}\ \mathsf{Cut}}{\Gamma_1, \Gamma_2, \Delta \vdash C}\ \mathsf{Cut}$$

and show that $h_1 = h_2$.

b) Suppose now $\mathcal{C}$ has *also* binary products, given by $\times$. Given that the sequents $\Gamma \vdash A$, $\Gamma \vdash B$ and $A, \Delta \vdash C$ are provable, and that their interpretations in $\mathcal{C}$ are $f_1 : \Gamma \to A$, $f_2 : \Gamma \to B$ and $g : A \otimes \Delta \to C$ respectively, find the interpretations h_1, h_2 of the following proofs:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash A} \quad \dfrac{\vdots}{\Gamma \vdash B}}{\Gamma \vdash A \,\&\, B}\ \&\mathsf{R} \qquad \dfrac{\dfrac{\vdots}{A, \Delta \vdash C}}{A \,\&\, B, \Delta \vdash C}\ \&\mathsf{L}}{\Gamma, \Delta \vdash C}\ \mathsf{Cut} \qquad\qquad \dfrac{\dfrac{\vdots}{\Gamma \vdash A} \quad \dfrac{\vdots}{A, \Delta \vdash C}}{\Gamma, \Delta \vdash C}\ \mathsf{Cut}$$

and show that $h_1 = h_2$.

3. Show that the condition $l_I = r_I$ in the definition of monoidal categories is redundant.
Moreover, show that the condition $\mathsf{id}_A \otimes l_B = a_{A,I,B} \circ r_A \otimes \mathsf{id}_B$ in the definition of symmetric monoidal categories is redundant.

1.8 Monads and comonads

1.8.1 Basics

Definition 49 A ***monad*** over a category $\mathcal{C}$ is a triple (T, η, μ) where T is an endofunctor on $\mathcal{C}$ and $\eta : \mathsf{Id}_{\mathcal{C}} \to T$, $\mu : T^2 \to T$ are natural transformations such that the following diagrams commute.

$$\begin{array}{ccc} T^3A & \xrightarrow{\mu_{TA}} & T^2A \\ {\scriptstyle T\mu_A}\downarrow & & \downarrow{\scriptstyle \mu_A} \\ T^2A & \xrightarrow[\mu_A]{} & TA \end{array} \qquad\qquad \begin{array}{ccc} TA & \xrightarrow{\eta_{TA}} & T^2A \\ {\scriptstyle T\eta_A}\downarrow & \searrow^{\mathsf{id}_{TA}} & \downarrow{\scriptstyle \mu_A} \\ T^2A & \xrightarrow[\mu_A]{} & TA \end{array}$$

▲

We usually call η the *unit* of the monad, and μ its *multiplication*; the whole terminology comes from monoids. Let us now proceed to some examples.

- Let $\mathcal{C}$ be a category with coproducts and let E be an object in $\mathcal{C}$. We can define a monad (T, η, μ) of E-coproducts (computationally, E-exceptions) by taking $T : \mathcal{C} \to \mathcal{C}$ to be the functor $\_ + E$, and η, μ as follows.

$$T := A \mapsto A + E\,,\ f \mapsto f + \mathsf{id}_E$$
$$\eta_A := A \xrightarrow{\mathsf{in}_1} A + E$$
$$\mu_A := (A + E) + E \xrightarrow{[\mathsf{id}_{A+E},\, \mathsf{in}_2]} A + E$$

That η, μ are natural transformations follows from the fact that injections and copairings are. Moreover, the monadic diagrams follow from the properties of the coproduct. For example,

$$\begin{aligned}
\mu_A \circ \mu_{TA} &= [\mu_A \circ \mu_{TA} \circ \mathsf{in}_1, \mu_A \circ \mu_{TA} \circ \mathsf{in}_2] \\
&= [\mu_A \circ [\mathsf{id}_{TA+E}, \mathsf{in}_2] \circ \mathsf{in}_1, \mu_A \circ [\mathsf{id}_{TA+E}, \mathsf{in}_2] \circ \mathsf{in}_2] \\
&= [\mu_A \circ \mathsf{id}_{TA+E}, \mu_A \circ \mathsf{in}_2] = [\mu_A, [\mathsf{id}_{A+E}, \mathsf{in}_2] \circ \mathsf{in}_2] \\
&= [\mu_A, \mathsf{in}_2] = [\mathsf{id}_{A+E} \circ \mu_A, \mathsf{in}_2 \circ \mathsf{id}_E] = [\mathsf{id}_{A+E}, \mathsf{in}_2] \circ (\mu_A + \mathsf{id}_E) \\
&= \mu_A \circ T\mu_A \,.
\end{aligned}$$

- Now let $\mathcal{C}$ be a cartesian closed category and let ξ be some object in $\mathcal{C}$. We can define a monad of ξ-side-effects by taking T to be the functor $\xi \Rightarrow (\_ \times \xi)$, and η, μ as follows (recall Λ and ev for CCC's).

$$\begin{aligned}
T &:= A \mapsto \xi \Rightarrow (A \times \xi)\,,\; f \mapsto \xi \Rightarrow (f \times \mathsf{id}_\xi) \\
\eta_A &:= \Lambda(\, A \times \xi \xrightarrow{\mathsf{id}_{A\times\xi}} A \times \xi \,) \\
\mu_A &:= \Lambda(\, T(TA) \times \xi \xrightarrow{\mathsf{ev}_{\xi,TA\times\xi}} TA \times \xi \xrightarrow{\mathsf{ev}_{\xi,A\times\xi}} A \times \xi \,)
\end{aligned}$$

Naturality of η, μ follows from naturality of Λ: for any $f : A \to A'$,

$$\begin{aligned}
Tf \circ \eta_A &= (\xi \Rightarrow f \times \mathsf{id}_\xi) \circ \Lambda(\mathsf{id}_{A\times\xi}) = \Lambda(f \times \mathsf{id}_\xi \circ \mathsf{id}_{A\times\xi}) \\
&= \Lambda(\mathsf{id}_{A'\times\xi} \circ f \times \mathsf{id}_\xi) = \Lambda(\mathsf{id}_{A'\times\xi}) \circ f = \eta_{A'} \circ f \,,
\end{aligned}$$

$$\begin{aligned}
\mu_{A'} \circ T^2 f &= \Lambda(\mathsf{ev}_{\xi,A'\times\xi} \circ \mathsf{ev}_{\xi,TA'\times\xi}) \circ T^2 f = \Lambda(\mathsf{ev}_{\xi,A'\times\xi} \circ \mathsf{ev}_{\xi,TA'\times\xi} \circ T^2 f \times \mathsf{id}_\xi) \\
&= \Lambda(\mathsf{ev}_{\xi,A'\times\xi} \circ Tf \times \mathsf{id}_\xi \circ \mathsf{ev}_{\xi,TA\times\xi}) = \Lambda(f \times \mathsf{id}_\xi \circ \mathsf{ev}_{\xi,A\times\xi} \circ \mathsf{ev}_{\xi,TA\times\xi}) \\
&= (\xi \Rightarrow f \times \mathsf{id}_\xi) \circ \Lambda(\mathsf{ev}_{\xi,A\times\xi} \circ \mathsf{ev}_{\xi,TA\times\xi}) = Tf \circ \mu_A \,.
\end{aligned}$$

The monadic properties are shown in a similar manner.

- Our third example employs the functor $U : \mathbf{Mon} \to \mathbf{Set}$. In particular, we take $T := U \circ \mathsf{MList}$ and η, μ as follows.

$$\begin{aligned}
T := X &\mapsto \bigcup\nolimits_{n\in\omega} \{[x_1, \ldots, x_n] \mid x_1, \ldots, x_n \in X\}\,, \\
f &\mapsto (\, [x_1, \ldots, x_n] \mapsto [f(x_1), \ldots, f(x_n)]\,)\,.
\end{aligned}$$

$$\eta_X := x \mapsto [x]$$

$$\mu_X := [[x_{11}, \ldots, x_{1n_1}], \ldots, [x_{k1}, \ldots, x_{kn_k}]] \mapsto [x_{11}, \ldots, x_{1n_1}, \ldots, x_{k1}, \ldots, x_{kn_k}]$$

Naturality of η, μ is obvious — besides, η is the unit of the corresponding adjunction. The monadic diagrams are also straightforward: they correspond to the following equalities of mappings (we use $[\mathbf{x}]$ for $[x_1, \ldots, x_n]$).

$$\begin{array}{ccc}
[[[\mathbf{x}_{11}], ..., [\mathbf{x}_{1n_1}]], ..., [[\mathbf{x}_{k1}], ..., [\mathbf{x}_{kn_k}]]] & \overset{\mu}{\longmapsto} & [[\mathbf{x}_{11}], ..., [\mathbf{x}_{1n_1}], ..., [\mathbf{x}_{k1}], ..., [\mathbf{x}_{kn_k}]] \\
\downarrow{\scriptstyle T\mu} & & \downarrow{\scriptstyle \mu} \\
[[\mathbf{x}_{11}, ..., \mathbf{x}_{1n_1}], \ldots, [\mathbf{x}_{k1}, ..., \mathbf{x}_{kn_k}]] & \underset{\mu}{\longmapsto} & [\mathbf{x}_{11}, ..., \mathbf{x}_{1n_1}, \ldots, \mathbf{x}_{k1}, ..., \mathbf{x}_{kn_k}]
\end{array}$$

$$\begin{array}{ccc} [x_1, \dots, x_n] & \overset{\eta}{\longmapsto} & [[x_1, \dots, x_n]] \\ {\scriptstyle T\eta}\downarrow & \overset{\mathsf{id}}{\searrow} & \downarrow{\scriptstyle \mu} \\ [[x_1], \dots, [x_n]] & \underset{\mu}{\longmapsto} & [x_1, \dots, x_n] \end{array}$$

Exercise 47. Show that the E-coproduct monad and the ξ-side-effect monads are indeed monads.

Our discussion on monads can be dualised, leading us to *comonads*.

Definition 50 A ***comonad*** over a category $\mathcal{C}$ is a triple (Q, ε, δ) where Q is an endofunctor on $\mathcal{C}$ and $\varepsilon : Q \to \mathsf{Id}_{\mathcal{C}}$, $\delta : Q \to Q^2$ are natural transformations such that the following diagrams commute.

$$\begin{array}{ccc} QA & \xrightarrow{\delta_A} & Q^2A \\ {\scriptstyle \delta_A}\downarrow & & \downarrow{\scriptstyle \delta_{QA}} \\ Q^2A & \xrightarrow[Q\delta_A]{} & Q^3A \end{array} \qquad \begin{array}{ccc} QA & \xrightarrow{\delta_A} & Q^2A \\ {\scriptstyle \delta_A}\downarrow & \overset{\mathsf{id}_{QA}}{\searrow} & \downarrow{\scriptstyle \varepsilon_{QA}} \\ Q^2A & \xrightarrow[Q\varepsilon_A]{} & QA \end{array}$$

▲

ε is the *counit* of the comonad, and δ its *comultiplication*. Two of our examples from monads dualise to comonads.

- If $\mathcal{C}$ has finite products then, for any object S, we can define the S-product comonad with functor $Q := S \times \_$.
- We can form a comonad on **Mon** with functor $Q := \mathsf{MList} \circ U$ (and counit that of the corresponding adjunction).

Exercise 48. Give an explicit description of the comonad on **Mon** with functor $Q := \mathsf{MList} \circ U$ described above. Verify it is a comonad.

1.8.2 (Co)Monads of an adjunction

In the previous section, we saw that an adjunction between **Mon** and **Set** yielded a monad on **Set** (and a comonad on **Mon**), with its unit being the unit of the adjunction. We now show that this observation generalises to any adjunction. Recall that an adjunction is specified by:

- a pair of functors $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$,
- for each $A \in Ob(\mathcal{C}), B \in Ob(\mathcal{D})$, a bijection $\theta_{A,B} : \mathcal{C}(A, GB) \cong \mathcal{D}(FA, B)$ natural in A, B.

For such an adjunction we build a monad on $\mathcal{C}$: the functor of the monad is simply $T := G \circ F$, and unit and multiplication are defined by setting

$$\eta_A : A \longrightarrow GFA := \theta^{-1}_{A,FA}(\mathsf{id}_{FA}),$$
$$\mu_A : GFGFA \longrightarrow GFA := G(\theta_{GFA,FA}(\mathsf{id}_{GFA})).$$

Observe that η is the unit of the adjunction.

Proposition 14. *Let (F, G, η) be an adjunction. Then, the triple (T, η, μ) defined above is a monad on $\mathcal{C}$.*

Proof: Recall that naturality of θ means concretely that, for any $f : A \to GB$, $g : A' \to A$ and $h : B \to B'$,

$$\theta_{A',B'}(Gh \circ f \circ g) = h \circ \theta_{A,B}(f) \circ Fg .$$

Now, we first show naturality of μ:

$$\begin{aligned} GFGFf \circ \mu_B &= G\theta_{GFB,FB}(\mathsf{id}_{GFB}) \circ GFGFf = G(\theta_{GFB,FB}(\mathsf{id}_{GFB}) \circ FGFf) \\ &\overset{\text{nat.}\theta}{=} G\theta_{GFA,FB}(\mathsf{id}_{GFB} \circ GFf) = G\theta_{GFA,FB}(GFf \circ \mathsf{id}_{GFA}) \\ &\overset{\text{nat.}\theta}{=} G(Ff \circ \theta_{GFA,FA}(\mathsf{id}_{GFA})) = GFf \circ \mu_A . \end{aligned}$$

The monoidal condition for μ also follows from naturality of θ:

$$\begin{aligned} \mu_A \circ \mu_{GFA} &= G(\theta(\mathsf{id}_{GFA}) \circ \theta(\mathsf{id}_{GFGFA})) \overset{\text{nat}}{=} G\theta(G\theta(\mathsf{id}_{GFA}) \circ \mathsf{id}_{GFGFA}) \\ &= G\theta(\mathsf{id}_{GFA} \circ G\theta(\mathsf{id}_{GFA})) \overset{\text{nat}}{=} G(\theta(\mathsf{id}_{GFA}) \circ FG\theta(\mathsf{id}_{GFA})) \\ &= \mu_A \circ GF\mu_A . \end{aligned}$$

Finally, for the η-μ conditions we also use the universality diagram for η and the uniqueness property (in equational form).

$$\begin{aligned} \mu_A \circ \eta_{GFA} &= G\theta_{GFA,FA}(\mathsf{id}_{GFA}) \circ \eta_{GFA} = \mathsf{id}_{GFA} , \\ \mu_A \circ GF\eta_{GFA} &= G\theta_{GFA,FA}(\mathsf{id}_{GFA}) \circ GF\eta_{GFA} = G(\theta_{GFA,FA}(\mathsf{id}_{GFA}) \circ F\eta_{GFA}) \\ &\overset{\text{nat}}{=} G\theta(\mathsf{id}_{GFA} \circ \eta_{GFA}) = G\theta(G\mathsf{id}_{FA} \circ \eta_{GFA}) = G\mathsf{id}_{FA} = GFA . \end{aligned}$$

■

Hence, every adjunction gives rise to a monad. It turns out that the converse is also true: *every* monad is described by means of an adjunction in this way. In particular, there are two canonical constructions of adjunctions from a given monad: the *Kleisli construction*, and the *Eilenberg-Moore construction*. These are in a sense minimal and maximal solutions to describing a monad via an adjunction. We describe the former one in the next section.

Finally, note that — because of the symmetric definition of adjunctions — the whole discussion can be dualised to comonads. That is, every adjunction gives rise to a comonad with counit that of the adjunction, and also every comonad can be derived from an adjunction in this manner.

1.8.3 The Kleisli construction

The Kleisli construction starts from a monad (T, η, μ) on a category $\mathcal{C}$ and builds a category $\mathcal{C}_T$ of T-computations, as follows.

Definition 51 Let (T, η, μ) be a monad on a category $\mathcal{C}$. Construct the ***Kleisli category*** $\mathcal{C}_T$ by taking the same objects as $\mathcal{C}$, and by including an arrow $f_{.T} : A \to B$ in $\mathcal{C}_T$ for each $f : A \to TB$ in $\mathcal{C}$. That is,

$$\begin{aligned} Ob(\mathcal{C}_T) &:= Ob(\mathcal{C})\,, \\ \mathcal{C}_T(A, B) &:= \{f_{.T} \mid f \in \mathcal{C}(A, TB)\}\,. \end{aligned}$$

The identity arrow for A in $\mathcal{C}_T$ is $\eta_{A.T}$, while the composite of $f_{.T} : A \to B$ and $g_{.T} : B \to C$ is $h_{.T}$, where:

$$h := A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu_C} TC\,.$$

▲

The conditions for $\mathcal{C}_T$ being a category correspond to the monadic conditions. For composition with identity, for any $f : A \to TB$,

$$\begin{aligned} f_{.T} \circ \eta_{A.T} &= (\mu_B \circ Tf \circ \eta_A)_{.T} = (\mu_B \circ \eta_B \circ f)_{.T} = f_{.T}\,, \\ \eta_{B.T} \circ f_{.T} &= (\mu_B \circ T\eta_B \circ f)_{.T} = f_{.T}\,. \end{aligned}$$

For associativity of composition, for any $f : A \to TB$, $g : B \to TC$ and $h : C \to TD$,

$$\begin{aligned} (h_{.T} \circ g_{.T}) \circ f_{.T} &= (\mu_D \circ Th \circ g)_{.T} \circ f_{.T} = (\mu_D \circ T(\mu_D \circ Th \circ g) \circ f)_{.T} \\ &= (\mu_D \circ T\mu_D \circ T^2h \circ Tg \circ f)_{.T} = (\mu_D \circ \mu_D \circ T^2h \circ Tg \circ f)_{.T} \\ &= (\mu_D \circ Th \circ \mu_C \circ Tg \circ f)_{.T} = h_{.T} \circ (g_{.T} \circ f_{.T})\,. \end{aligned}$$

Let us now proceed to build the adjunction between $\mathcal{C}$ and $\mathcal{C}_T$ that will eventually give us back the monad T. Construct the functors $F : \mathcal{C} \to \mathcal{C}_T$ and $G : \mathcal{C}_T \to \mathcal{C}$ as follows.

$$\begin{aligned} F &:= A \mapsto A\,,\ (f : A \to B) \mapsto ((\eta_B \circ f)_{.T} : A \to B)\,, \\ G &:= A \mapsto TA\,,\ (f_{.T} : A \to B) \mapsto (\mu_B \circ Tf : TA \to TB)\,. \end{aligned}$$

Moreover, for each $A, B \in Ob(\mathcal{C})$, construct the following bijection of arrows.

$$\theta_{A,B} : \mathcal{C}(A, TB) \xrightarrow{\cong} \mathcal{C}_T(A, B) := f \mapsto f_{.T}$$

To establish that (F, G, θ) is an adjunction we need only show that θ is natural in A, B. So take $f : A \to TB$, $g : A' \to A$ and $h_{.T} : B \to B'$. We then have:

$$\begin{aligned}\theta_{A',B'}(Gh \circ f \circ g) = \theta_{A',B'}(\mu_{B'} \circ Th \circ f \circ g) &= (\mu_{B'} \circ Th \circ f \circ g)_{.T} \\ &= h_{.T} \circ (f \circ g)_{.T} = h_{.T} \circ (\mu_B \circ Tf \circ \eta_A \circ g)_{.T} \\ &= h_{.T} \circ f_{.T} \circ (\eta_A \circ g)_{.T} = h_{.T} \circ \theta_{A,B}(f) \circ Fg\,.\end{aligned}$$

The final step in this section is to verify that the monad (T', η', μ') that arises from this adjunction is the one we started from. The construction of T' follows the recipe given in the previous section, that is:

- $T' : \mathcal{C} \to \mathcal{C} := G \circ F$. Thus, T' maps each object A to TA, and each arrow $f : A \to B$ to $\mu_B \circ T\eta_A \circ Tf = Tf$.
- $\eta'_A : A \to TA := \theta^{-1}_{A,FA}(\mathsf{id}^{(\mathcal{C}_T)}_{FA}) = \theta^{-1}(\eta_{A.T}) = \eta_A$.
- $\mu'_A : T^2A \to TA := G\theta_{GFA,FA}(\mathsf{id}^{(\mathcal{C})}_{GFA}) = G\theta(\mathsf{id}_{TA}) = \mu_A \circ T\mathsf{id}_{TA} = \mu_A$.

Thus, we have indeed obtained the initial (T, η, μ).

The Kleisli construction on a comonad

Dually to the Kleisli category of a monad we can construct the Kleisli category of a comonad[4] — and reobtain the comonad through an adjunction between the Kleisli category and the original one. Specifically, given a category $\mathcal{C}$ and a comonad (Q, ε, δ) on $\mathcal{C}$, we define the category $\mathcal{C}_Q$ as follows.

$$\begin{aligned}Ob(\mathcal{C}_Q) &:= Ob(\mathcal{C}) \\ \mathcal{C}_Q(A, B) &:= \{f_{.Q} \mid f \in \mathcal{C}(QA, B)\} \\ \mathsf{id}^{(\mathcal{C}_Q)}_A &:= \varepsilon_{A.Q} \\ g_{.Q} \circ f_{.Q} &:= (g \circ Qf \circ \delta_A)_{.Q}\end{aligned}$$

The Kleisli category of a comonad will be of use in the next sections, where comonads will be considered for modeling *bang* of Linear Logic. We end this section by showing a result that will be of use then.

Proposition 15. *Let $\mathcal{C}$ be a category and (Q, ε, δ) be a comonad on $\mathcal{C}$. If $\mathcal{C}$ has binary products then so does $\mathcal{C}_Q$.*

Proof: Let A, B be objects in $\mathcal{C}, \mathcal{C}_Q$. We claim that their product in $\mathcal{C}_Q$ is given by $(A \times B, p_1, p_2)$, where

$$p_1 := \big(Q(A \times B) \xrightarrow{\varepsilon} A \times B \xrightarrow{\pi_1} A\big)_{.Q}$$

and p_2 is similar. Now, for each $f_{.Q} : C \to A$ and $g_{.Q} : C \to B$, setting $\langle f_{.Q}, g_{.Q}\rangle := \langle f, g\rangle_{.Q}$ we have:

$$p_1 \circ \langle f_{.Q}, g_{.Q}\rangle = (\pi_1 \circ \varepsilon \circ Q\langle f, g\rangle \circ \delta)_{.Q} = (\pi_1 \circ \langle f, g\rangle \circ \varepsilon \circ \delta)_{.Q} = f_{.Q}\,,$$

and similarly $p_2 \circ \langle f_{.Q}, g_{.Q}\rangle = g_{.Q}$. Finally, for any $h_{.Q} : C \to A \times B$,

$$\begin{aligned}\langle p_1 \circ h_{.Q}, p_2 \circ h_{.Q}\rangle &= \langle \pi_1 \circ \varepsilon \circ Qh \circ \delta, \pi_2 \circ \varepsilon \circ Qh \circ \delta\rangle_{.Q} = \langle \pi_1 \circ h, \pi_2 \circ h\rangle_{.Q} \\ &= h_{.Q}\,.\end{aligned}$$

∎

[4] in some texts, this is called a *coKleisli* category.

Exercise 49. Show that the Kleisli category $\mathcal{C}_Q$ of a comonad (Q, ε, δ) has a terminal object when $\mathcal{C}$ does.

1.8.4 Modeling of Linear exponentials

In this section we employ comonads in order to model the exponential *bang* operator, $\mathbf{!}$, of Linear Logic. Let us start with modeling a *weak bang* operator, $\hat{\mathbf{!}}$, which involves solely the following proof rules.

$$\dfrac{\Gamma, A \vdash B}{\Gamma, \hat{!}A \vdash B}\,\hat{!}\mathsf{L} \qquad\qquad \dfrac{\hat{!}B \vdash A}{\hat{!}B \vdash \hat{!}A}\,\hat{!}\mathsf{R}$$

Observe that, compared to $\mathbf{!}$, $\hat{\mathbf{!}}$ is weak in its Right rule.

Let us now assume as given a symmetric monoidal category $\mathcal{C}$ along with a comonad (Q, ε, δ) on $\mathcal{C}$. As seen previously, $\mathcal{C}$ is a model of $(\otimes\!\multimap)$-Linear Logic. Moreover, $(\mathcal{C}, Q)$ yields a model of $(\otimes\!\multimap\!\hat{!})$-Linear Logic by modeling each formula $\hat{!}A$ by QA (i.e. Q applied to the translation of A). The rules for weak bang are then interpreted as follows.

$$\dfrac{f : \Gamma \otimes A \longrightarrow B}{f \circ \mathsf{id}_\Gamma \otimes \varepsilon_A : \Gamma \otimes QA \longrightarrow B} \qquad\qquad \dfrac{f : QB \longrightarrow A}{Qf \circ \delta_B : QB \longrightarrow QA}$$

We know that arrow-equalities in the translation correspond to proof-transformations in the proof system. Let us see how this applies to the comonadic laws for ε. Firstly, naturality of ε, which is expressed by $\varepsilon_B \circ Qf = f \circ \varepsilon_A$, for each $f : A \to B$, corresponds to the following situation. Given a proof of the sequent $A \vdash B$, the following proof-transformation is valid.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{A \vdash B}}{\hat{!}A \vdash B}\,\hat{!}\mathsf{L}}{\hat{!}A \vdash \hat{!}B}\,\hat{!}\mathsf{R} \qquad \dfrac{\dfrac{}{B \vdash B}}{\hat{!}B \vdash B}\,\hat{!}\mathsf{L}}{\hat{!}A \vdash B}\,\mathsf{Cut} \quad\Longrightarrow\quad \dfrac{\dfrac{\vdots}{A \vdash B}}{\hat{!}A \vdash B}\,\hat{!}\mathsf{L}$$

Moreover, the comonadic law $\varepsilon_{QA} \circ \delta_A = \mathsf{id}_{QA} = Q\varepsilon_A \circ \delta_A$ corresponds to the following transformations.

$$\dfrac{\dfrac{\dfrac{}{\hat{!}A \vdash \hat{!}A}\,\mathsf{Id}}{\hat{!}A \vdash \hat{!}\hat{!}A}\,\hat{!}\mathsf{R} \qquad \dfrac{\dfrac{}{\hat{!}A \vdash \hat{!}A}\,\mathsf{Id}}{\hat{!}\hat{!}A \vdash \hat{!}A}\,\hat{!}\mathsf{L}}{\hat{!}A \vdash \hat{!}A}\,\mathsf{Cut} \quad\Longrightarrow\quad \dfrac{}{\hat{!}A \vdash \hat{!}A}\,\mathsf{Id} \quad\Longleftarrow\quad \dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Id}}{\hat{!}A \vdash A}\,\hat{!}\mathsf{L}}{\hat{!}A \vdash \hat{!}A}\,\hat{!}\mathsf{R}$$

Exercise 50. Find the proof-transformations corresponding to naturality of δ and to the comonadic law $\delta_{QA} \circ \delta_A = Q\delta_A \circ \delta_A$.

In order to extend our translation to the general !R rule, we need arrows in $\mathcal{C}$ of the form

$$Q^2A_1 \otimes \cdots \otimes Q^2A_n \longrightarrow Q(QA_1 \otimes \cdots \otimes QA_n)\,.$$

Hence, we need to impose (a coherent) distributivity of the tensor — either binary ($\otimes$) or nullary (I) — over the comonad Q. This can be formalised by stipulating that Q be a *symmetric monoidal* endofunctor.

Definition 52 Let $(\mathcal{C}, \otimes, I, a, l, r, s)$ and $(\mathcal{C}', \otimes', I', a', l', r', s')$ be symmetric monoidal categories. A functor $F : \mathcal{C} \to \mathcal{D}$ is called ***symmetric monoidal*** if there exist:

- a morphism $m_0 : I' \to F(I)$,
- a natural transformation $m_2 : F(\_) \otimes' F(\_) \to F(\_ \otimes \_)$,

such that the following diagrams commute.

$$\begin{array}{ccccc}
FA \otimes' (FB \otimes' FC) & \xrightarrow{\mathsf{id} \times m_2} & FA \otimes' F(B \otimes C) & \xrightarrow{m_2} & F(A \otimes (B \otimes C)) \\
{\scriptstyle a'}\downarrow & & & & \downarrow{\scriptstyle Fa} \\
(FA \otimes' FB) \otimes' FC & \xrightarrow[m_2 \times \mathsf{id}]{} & F(A \otimes B) \otimes' FC & \xrightarrow[m_2]{} & F((A \otimes B) \otimes C)
\end{array}$$

$$\begin{array}{ccc}
FA \otimes' I' & \xrightarrow{\mathsf{id} \otimes' m_0} & FA \otimes' FI \\
{\scriptstyle r'}\downarrow & & \downarrow{\scriptstyle m_2} \\
FA & \xleftarrow[Fr]{} & F(A \otimes I)
\end{array}
\qquad
\begin{array}{ccc}
FA \otimes' FB & \xrightarrow{m_2} & F(A \otimes B) \\
{\scriptstyle s}\downarrow & & \downarrow{\scriptstyle Fs} \\
FB \otimes' FA & \xrightarrow[m_2]{} & F(B \otimes A)
\end{array}$$

We may write such an F as (F, m). Moreover, if $(F, m), (G, n) : \mathcal{C} \to \mathcal{C}'$ are (symmetric) monoidal functors then a natural transformation $\phi : F \to G$ is called ***monoidal*** whenever the following diagrams commute.

$$\begin{array}{ccc}
I' & \xrightarrow{m_0} & FI \\
 & {\scriptstyle n_0}\searrow & \downarrow{\scriptstyle \phi} \\
 & & GI
\end{array}
\qquad
\begin{array}{ccc}
FA \otimes' FB & \xrightarrow{m_2} & F(A \otimes B) \\
{\scriptstyle \phi \otimes \phi}\downarrow & & \downarrow{\scriptstyle \phi} \\
GA \otimes' GB & \xrightarrow[n_2]{} & G(A \otimes B)
\end{array}$$

▲

For example, the identity functor is symmetric monoidal. Moreover, if F and G are symmetric monoidal functors then so is $G \circ F$. Other examples are the following.

- The constant endofunctor K_I, which maps each object to I and each arrow to id_I, is symmetric monoidal with structure maps:

$$m_0 : I \longrightarrow I := \mathsf{id}_I\,, \quad m_2 : I \otimes I \longrightarrow I := r_I\,.$$

- The endofunctor $\otimes \circ \langle \mathsf{Id}_{\mathcal{C}}, \mathsf{Id}_{\mathcal{C}} \rangle$, which maps each object A to $A \otimes A$ and each arrow f to $f \otimes f$, is symmetric monoidal with:

$$m_0 : I \longrightarrow I \otimes I := r_I^{-1}\,, \quad m_2 := (A \otimes A) \otimes (B \otimes B) \longrightarrow (A \otimes B) \otimes (A \otimes B)\,,$$

the latter given by use of structural transformations.

Exercise 51. Verify that if $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{D} \to \mathcal{E}$ are symmetric monoidal functors then so is $G \circ F$.

Definition 53 A comonad (Q, ε, δ) on a SMCC $\mathcal{C}$ is called a ***monoidal comonad*** if Q is a symmetric monoidal functor, say (Q, m), and ε, δ are monoidal natural transformations. We write Q as $(Q, \varepsilon, \delta, m)$. ▲

Now let us assume $\mathcal{C}$ is a SMCC and $(Q, \varepsilon, \delta, m)$ is a monoidal comonad on $\mathcal{C}$. The coherence of m_2 with a, expressed by the first diagram of symmetric monoidal functors, allows us to generalise m_0 and m_2 to arbitrary arities and assume arrows:

$$m_n : QA_1 \otimes \cdots \otimes QA_n \longrightarrow Q(A_1 \otimes \cdots \otimes A_n)\,.$$

We can give the interpretation of the Right rule for bang as follows.

$$\frac{f : QB_1 \otimes \cdots \otimes QB_n \longrightarrow A}{Qf \circ m_n \circ (\delta_{B_1} \otimes \cdots \otimes \delta_{B_n}) : QB_1 \otimes \cdots \otimes QB_n \longrightarrow QA}$$

Contraction and Weakening

Our discussion on the categorical modeling of linear exponentials has only touched the issues of Right and Left rules. However, this is only the beginning of the story: we also need adequate structure for translating Contraction and Weakening.

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ \mathsf{Contr} \qquad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ \mathsf{Weak}$$

The solution is to use appropriate (monoidal) natural transformations. For Contraction, we stipulate a transformation with components $d_A : QA \to QA \otimes QA$, i.e.

$$d : Q \longrightarrow \otimes \circ \langle Q, Q \rangle\,.$$

For Weakening, a transformation with components $e_A : QA \to I$, i.e.

$$e : Q \longrightarrow K_I.$$

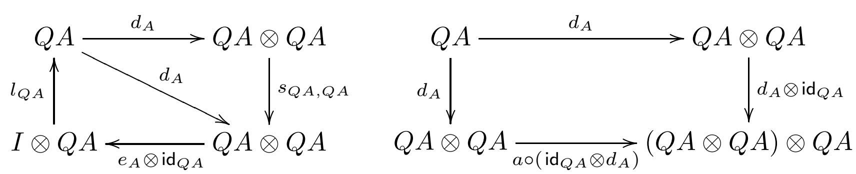
Although the above allow the categorical interpretation of the proof-rules, they do not necessarily preserve the intended proof-transformations. For that, we need to impose some further coherence conditions, which are epitomised in the following notion.

Definition 54 Let $\mathcal{C}$ be a SMCC. A monoidal comonad $(Q, \varepsilon, \delta, m)$ on $\mathcal{C}$ is called a ***linear exponential comonad*** if there exist monoidal natural transformations

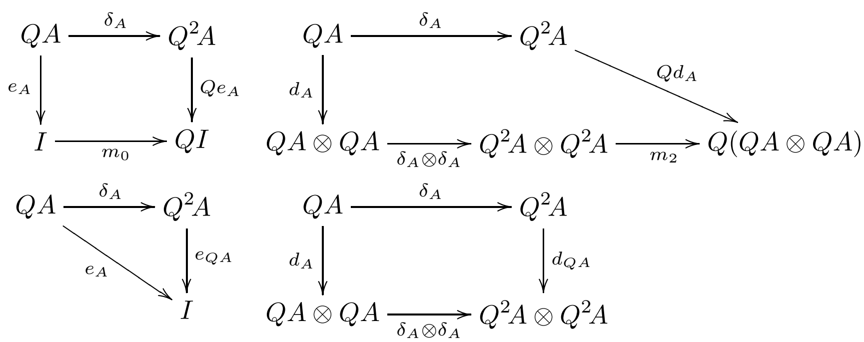
$$d : Q \longrightarrow \otimes \circ \langle Q, Q \rangle\,, \quad e : Q \longrightarrow K_I,$$

such that:

(a) for each object A, the triple (QA, d_A, e_A) is a *commutative comonoid in* $\mathcal{C}$, i.e. the following diagrams commute,



(b) for each object A, the following diagrams commute.



We write Q as $(Q, \varepsilon, \delta, m, d, e)$. ▲

Exercise 52. Give the categorical interpretation of Contraction and Weakening in a SMCC $\mathcal{C}$ with a linear exponential comonad.

1.8.5 Including products

We now consider the fragment of Linear Logic which includes all four linear connectives we have seen thus far, i.e. $\otimes \multimap \mathbf{!}\&$, and their respective proof rules (see definitions 45, 46). The categorical modeling of $(\otimes \multimap \mathbf{!}\&)$-Linear Logic requires:

- a symmetric monoidal closed category $\mathcal{C}$,
- a linear exponential comonad $(Q, \varepsilon, \delta, m, d, e)$ on $\mathcal{C}$,
- finite products in $\mathcal{C}$.

The above structure is adequate for modeling the proof rules as we have seen previously. Moreover, it provides rich structure for the Kleisli category $\mathcal{C}_Q$. The next result, and its proof, demonstrates categorically the 'interpretation' of ordinary logic within Linear Logic given by:

$$A \Rightarrow B \equiv \mathbf{!}A \multimap B\,.$$

Proposition 16. *Let $\mathcal{C}$ be a SMCC with finite products and let $(Q, \varepsilon, \delta, m, d, e)$ be a linear exponential comonad on $\mathcal{C}$. Then:*

(a) *The Kleisli category $\mathcal{C}_Q$ has finite products.*
(b) *There exists an isomorphism $i : Q\mathbf{1} \to I$ and a natural isomorphism $j : Q(\_ \times \_) \to Q(\_) \otimes Q(\_)$.*
(c) *$\mathcal{C}_Q$ is cartesian closed.*

Proof: Part (a) has been shown previously (proposition 15, exercise 49), and part (b) is left as exercise. For (c), we have the following isomorphisms:

$$\begin{aligned}
\mathcal{C}_Q(A \times B, C) &= \mathcal{C}(Q(A \times B), C) && \text{definition of } \mathcal{C}_Q \\
&\cong \mathcal{C}(QA \otimes QB, C) && \text{part (b)} \\
&\cong \mathcal{C}(QA, QB \multimap C) && \text{monoidal closure of } \mathcal{C} \\
&= \mathcal{C}_Q(A, QB \multimap C) && \text{defn of } \mathcal{C}_Q.
\end{aligned}$$

Concretely, we obtain $\theta_A : \mathcal{C}_Q(A \times B, C) \xrightarrow{\cong} \mathcal{C}_Q(A, QB \multimap C)$ by:

$$\begin{aligned}
\theta_A &:= (f_{.Q} : A \times B \to C) \longmapsto (\Lambda(f \circ j^{-1}_{A,B}))_{.Q} \\
\theta_A^{-1} &:= (g_{.Q} : A \to QB \multimap C) \longmapsto (\Lambda^{-1}(g \circ j))_{.Q}\,.
\end{aligned}$$

Clearly, θ_A is a bijection. In order to establish couniversality of the exponential, we need also show naturality in A. So take $f_{.Q} : A \times B \to C$ and $h_{.Q} : A' \to A$. Note first that the following commutes.

$$\begin{array}{ccccc}
Q(A \times B) & \xrightarrow{\delta} & Q^2(A \times B) & \xrightarrow{Q\langle Q\pi_1, Q\pi_2\rangle} & Q(QA \times QB) \\
\downarrow{\scriptstyle j} & & & & \downarrow{\scriptstyle j} \\
QA \otimes QB & & \xrightarrow[\delta\otimes\delta]{} & & Q^2A \otimes Q^2B
\end{array} \qquad (*)$$

Note also that, for any $h_{i.Q} : A'_i \to A_i$ in $\mathcal{C}_Q$, we have:

$$h_{1.Q} \times h_{2.Q} := \big(Q(A'_1 \times A'_2) \xrightarrow{\langle Q\pi_1, Q\pi_2\rangle} QA'_1 \times QA'_2 \xrightarrow{h_1 \times h_2} A_1 \times A_2\big)_{.Q}$$

Thus, noting that $\mathsf{id}_B^{(\mathcal{C}_Q)} = \varepsilon_{B.Q}$,

$$\begin{aligned}
\theta_{A'}(f_{.Q} \circ h_{.Q} \times \varepsilon_{.Q}) &= \big(\Lambda(f \circ Q(h \times \varepsilon \circ \langle Q\pi_1, Q\pi_2\rangle) \circ \delta \circ j^{-1})\big)_{.Q} \\
&= \big(\Lambda(f \circ Q(h \times \varepsilon) \circ Q\langle Q\pi_1, Q\pi_2\rangle \circ \delta \circ j^{-1})\big)_{.Q} \\
&\overset{(*)}{=} \big(\Lambda(f \circ Q(h \times \varepsilon) \circ j^{-1} \circ \delta \otimes \delta)\big)_{.Q} \\
&= \big(\Lambda(f \circ j^{-1} \circ Qh \otimes Q\varepsilon \circ \delta \otimes \delta)\big)_{.Q} \\
&= \big(\Lambda(f \circ j^{-1} \circ (Qh \circ \delta) \otimes \mathsf{id})\big)_{.Q} \\
&= \big(\Lambda(f \circ j^{-1}) \circ Qh \circ \delta\big)_{.Q} = \theta_A(f_{.Q}) \circ h_{.Q}
\end{aligned}$$

as required. Hence, $\mathcal{C}_Q$ has exponentials and is therefore a CCC. In particular, the exponential of objects B, C is $QB \multimap C$. ■

Exercise 53. Show part (b) of proposition 16. For the defined j, show commutativity of $(*)$.

1.8.6 Exercises

1. We say that a category $\mathcal{C}$ is *well-pointed* if it contains a terminal object $\mathbf{1}$ and, for any pair of arrows $f, g : A \to B$,

$$f \neq g \implies \exists h : 1 \longrightarrow A.\, f \circ h \neq g \circ h\,.$$

Let now $\mathcal{C}$ be a well-pointed category with a terminal object $\mathbf{1}$ and binary coproducts, and consider the functor $G : \mathcal{C} \to \mathcal{C}$ given by:

$$G := A \mapsto A + \mathbf{1}\,,\; f \mapsto f + \mathsf{id}_{\mathbf{1}}\,.$$

If $\mathcal{C}(\mathbf{1}, \mathbf{1}+\mathbf{1}) = \{\mathsf{in}_1, \mathsf{in}_2\}$ with $\mathsf{in}_1 \neq \mathsf{in}_2$, show that if (G, η, μ) is a monad on $\mathcal{C}$ then, for each object A:

$$\eta_A = A \xrightarrow{\mathsf{in}_1} A + \mathbf{1}\,, \quad \mu_A = (A + \mathbf{1}) + \mathbf{1} \xrightarrow{[\mathsf{id}_{A+\mathbf{1}}, \mathsf{in}_2]} A + \mathbf{1}\,.$$

2. Let $\mathcal{C}$ be a SMCC and let (Q, ε, δ) be a comonad on $\mathcal{C}$. Suppose that the sequents $\hat{!}A \vdash B$ and $\hat{!}B \vdash C$ are provable and let $f : FA \to B$ and $g : FB \to C$ be their interpretations (i.e. the interpretations of their proofs) in $\mathcal{C}$. Find the interpretations of the sequent $\hat{!}A \vdash \hat{!}C$ which correspond to each of the following proofs,

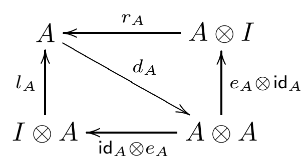
$$\dfrac{\dfrac{\dfrac{\vdots}{\hat{!}A \vdash B}}{\hat{!}A \vdash \hat{!}B}\,\hat{!}\mathsf{R} \qquad \dfrac{\dfrac{\vdots}{\hat{!}B \vdash C}}{\hat{!}B \vdash \hat{!}C}\,\hat{!}\mathsf{R}}{\hat{!}A \vdash \hat{!}C}\,\mathsf{Cut} \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\hat{!}A \vdash B}}{\hat{!}A \vdash \hat{!}B}\,\hat{!}\mathsf{R} \qquad \dfrac{\vdots}{\hat{!}B \vdash C}}{\hat{!}A \vdash C}\,\mathsf{Cut}}{\hat{!}A \vdash \hat{!}C}\,\hat{!}\mathsf{R}$$

and show that the two interpretations are equal.

3. Show that a symmetric monoidal category $\mathcal{C}$ has finite products (given by $\otimes, I$, etc.) iff there are monoidal natural transformations

$$d : \mathsf{Id}_{\mathcal{C}} \longrightarrow \otimes \circ \langle \mathsf{Id}_{\mathcal{C}}, \mathsf{Id}_{\mathcal{C}} \rangle\,, \quad e : \mathsf{Id}_{\mathcal{C}} \longrightarrow K_I\,,$$

such that the following diagram commutes, for any $A \in Ob(\mathcal{C})$.



A Review of Sets, Functions and Relations

Our aim in this Appendix is to provide a brief review of notions we will assume in the notes. If the first paragraph is not familiar to you, you will need to acquire more background before being ready to read the notes.

Cartesian products, relations and functions

Given sets X and Y, their cartesian product is

$$X \times Y = \{(x, y) \mid x \in X \;\wedge\; y \in Y\}\,.$$

A *relation* R from X to Y, written $R : X \to Y$, is a subset $R \subseteq X \times Y$. Given such a relation, we write $(x, y) \in R$, or equivalently $R(x, y)$. We compose relations as follows: if $R : X \to Y$ and $S : Y \to Z$, then for all $x \in X$ and $z \in Z$:

$$R; S(x, z) \equiv \exists y \in Y.\, R(x, y) \;\wedge\; S(y, z)\,.$$

A relation $f : X \to Y$ is a *function* if it satisfies the following two properties:

- (single-valuedness): if $(x, y) \in f$ and $(x, y') \in f$, then $y = y'$.
- (totality): for all $x \in X$, for some $y \in Y$, $(x, y) \in f$.

If f is a function, we write $f(x) = y$ or $f : x \mapsto y$ for $(x, y) \in f$. Function composition is written as follows: if $f : X \to Y$ and $g : Y \to Z$,

$$g \circ f(x) = g(f(x))\,.$$

It is easily checked that $g \circ f = f; g$, viewing functions as relations.

Equality of functions

Two functions $f, g : X \to Y$ are *equal* if they are equal as relations, *i.e.* as sets of ordered pairs. Equivalently, but more conveniently, we can write:

$$f = g \iff \forall x \in X.\, f(x) = g(x)\,.$$

The right-to-left implication is the standard tool for proving equality of functions on sets. As we shall see, when we enter the world of category theory, which takes a more general view of "arrows" $f : X \to Y$, for most purposes we have to leave this familiar tool behind!

Making the arrow notation for functions and relations unambiguous

Our definitions of functions and relations, as they stand, have an unfortunate ambiguity. Given a relation $R : X \to Y$, we cannot uniquely recover its "domain" X and "codomain" Y. In the case of a function, we can recover the domain, because of totality, but not the codomain.

Example Consider the set of ordered pairs $\{(n, n) \mid n \in \mathbb{N}\}$, where $\mathbb{N}$ is the set of natural numbers. Is this the identity function $\mathsf{id}_{\mathbb{N}} : \mathbb{N} \longrightarrow \mathbb{N}$, or the inclusion function $\mathsf{inc} : \mathbb{N} \hookrightarrow \mathbf{Z}$, where $\mathbb{Z}$ is the set of integers?

We wish to have unambiguous notions of domain and codomain for functions, and more generally relations. Thus we modify our official definition of a relation from X to Y to be an ordered triple (X, R, Y), where $R \subseteq X \times Y$. We then define composition of (X, R, Y) and (Y, S, Z) in the obvious fashion, as $(X, R; S, Z)$. We treat functions similarly. We shall not belabour this point in the notes, but it is implicit when we set up perhaps the most fundamental example of a category, namely the category of sets.

Size

We shall avoid explicit discussion of set-theoretical foundations in the text, but we include a few remarks for the interested reader. Occasionally, distinctions of set-theoretic size do matter in category theory. One example which does arise in the notes is when we consider **Cat**, the category of "all" categories. Does this category belong to (is it an object of) itself, at the risk of a Russell-type paradox? The way we avoid this is to impose some set-theoretic limitation of size on the categories gathered into **Cat**. **Cat** will then be too big to fit into itself. For example, we can limit **Cat** to those categories whose collections of objects and arrows form *sets* in the sense of some standard set theory such as ZFC. **Cat** will then be a proper class, and will not be an object of itself. One assumption we do make throughout the notes is that the categories we deal with are "locally small", *i.e.* that all hom-sets are indeed sets. Another place where some technical caveat would be in order is when we form functor categories. In practice, these issues never (well, hardly ever) cause problems, because of the strongly-typed nature of category theory. We leave the interested reader to delve further into these issues by consulting some of the standard texts.

B Guide to Further Reading

Of the many texts on category theory, we shall only mention a few, which may be particularly useful to someone who has read these notes and wishes to learn more.

The short text [10] is very nicely written and gently paced; it is probably a little easier going than these notes. Two texts which are written with a clarity and at a level which makes them ideal as a next step after these notes are [1] and [6]. Unfortunately, both are out of print.

Another very nicely written text, focussing on the connections between categories and logic, and especially topos theory, is [5], recently reissued by Dover Books. A classic text on categorical logic is [7].

A particularly useful feature of [3] is the large number of exercises with solutions.

The text [9] is a classic by one of the co-founders of category theory. It assumes considerable background knowledge of mathematics to fully appreciate its wide-ranging examples, but it provides invaluable coverage of the key topics.

The 3-volume handbook [4] provides coverage of a broad range of topics in category theory. The book [8] is somewhat idiosyncratic in style, but offers insights by one of the key contributors to category theory. We have not yet had the chance to study [2] in detail, but it looks useful, albeit exorbitantly priced.

References

1. M. Arbib and E. Manes, *Arrows, Structures and Functors: the Categorical Imperative*. Academic Press 1975.
2. S. Awodey, *Category Theory*, Oxford University Press 2006.
3. M. Barr and C. Wells, *Category Theory for Computing Science*, Third Edition, Publications CRM 1999.
4. F. Borceux, *Handbook of Categorical Algebra Volumes 1–3*, Cambridge University Press 1994.
5. R Goldblatt, *Topoi, the categorial analysis of logic*, North-Holland 1984. Reprinted by Dover Books 2006.
6. H. Herrlich and G. Strecker, *Category Theory*, Second Edition, Heldermann Verlag, 1979.
7. J. Lambek and P. J. Scott, *Introduction to higher-order categorical logic*, Cambridge University Press, 1986.
8. F. Lawvere and S. Schanuel, *Conceptual Mathematics: A First Introduction to Categories*, Cambridge University Press 1997.
9. S. Mac Lane, *Categories for the Working Mathematician*, Second Edition, Springer 1998.
10. B. Pierce, *Basic Category Theory for Computer Scientists*, MIT press 1991.