Introduction to Categorical Logic

[DRAFT: January 29, 2022]

Steve Awodey　Andrej Bauer

Contents

Chapter 1

Algebraic Theories

Algebraic theories are descriptions of structures that are given entirely in terms of operations and equations. All such algebraic notions have in common some quite deep and general properties, from the existence of free algebras to Lawvere's duality theory. The most basic of these are presented in this chapter. The development also serves as a first example and template for the general scheme of *functorial semantics*, to be applied to other logical notions in later chapters.

1.1 Syntax and semantics

We begin with a general approach to algebraic structures such as groups, rings, and lattices. These are characterized by axiomatizations which involve only a single sort of variables and constants, operations, and equations. It is important that the operations are defined everywhere, which excludes two important examples: fields, because the inverse of 0 is undefined, and categories because composition is defined only for certain pairs of morphisms.

Let us start with the quintessential algebraic theory: the theory of groups. In first-order logic, a group can be described as a set G with a binary operation $\cdot : G \times G \to G$, satisfying the two first-order axioms:

$$\forall\, x, y, z \in G \,.\, (x \cdot y) \cdot z = x \cdot (y \cdot z)$$
$$\exists\, e \in G \,.\, \forall\, x \in G \,.\, \exists\, y \in G \,.\, (e \cdot x = x \cdot e = x \wedge x \cdot y = y \cdot x = e)$$

Taking a closer look at the logical form of these axioms, we see that the second one, which expresses the existence of a unit and inverse elements, is somewhat unsatisfactory because it involves nested quantifiers. Not only does this complicate the interpretation, but it is not really necessary, since the unit element and inverse operation in a group are uniquely determined. Thus we can add them to the structure and reformulate as follows. The unit is to be represented by a distinguished *constant* $e \in G$, and the inverse is to be a unary *operation* ${}^{-1} : G \to G$. We then obtain an equivalent formulation in which all axioms can be expressed as (universally quantified) *equations*:

$$\begin{gathered} x \cdot (y \cdot z) = (x \cdot y) \cdot z \\ x \cdot e = x \qquad\qquad e \cdot x = x \\ x \cdot x^{-1} = e \qquad\qquad x^{-1} \cdot x = e \end{gathered}$$

The universal quantifiers $\forall\, x \in G, \forall\, y \in G$, etc. are no longer needed in stating the axioms, since we can interpret all variables as ranging over all elements of G (because of our restriction to totally defined operations). Nor do we really need to explicitly mention the particular set G in the specification. Finally, since the constant e can be regarded as a nullary operation, i.e., a function $e : \mathbf{1} \to G$, the specification of the group concept consists solely of operations and equations. This leads to the following general definition of an algebraic theory.

Definition 1.1.1. A *signature* Σ for an algebraic theory consists of a family of sets $\{\Sigma_k\}_{k \in \mathbb{N}}$. The elements of Σ_k are called the *k-ary operations.* In particular, the elements of Σ_0 are the *nullary operations* or *constants.*

The *terms* of a signature Σ are the expressions constructed inductively by the following rules:

1. variables x, y, z, ..., are terms,

2. if $t_1, \ldots, t_k$ are terms and $f \in \Sigma_k$ is a k-ary operation then $f(t_1, \ldots, t_k)$ is a term.

Definition 1.1.2 (cf. Definition **??**). An *algebraic theory* $\mathbb{T} = (\Sigma_{\mathbb{T}}, A_{\mathbb{T}})$ is given by a signature $\Sigma_{\mathbb{T}}$ and a set $A_{\mathbb{T}}$ of *axioms*, which are equations between terms (formally, pairs of terms).

Algebraic theories are also called *equational theories.* We do not assume that the sets Σ_k or $A_{\mathbb{T}}$ are finite, but the individual terms and equations always involve only finitely many variables.

Example 1.1.3. The theory of a commutative ring with unit is an algebraic theory. There are two nullary operations (constants) 0 and 1, a unary operation $-$, and two binary operations $+$ and $\cdot$. The equations are:

$$\begin{aligned} (x + y) + z &= x + (y + z) & (x \cdot y) \cdot z &= x \cdot (y \cdot z) \\ x + 0 &= x & x \cdot 1 &= x \\ 0 + x &= x & 1 \cdot x &= x \\ x + (-x) &= 0 & (x + y) \cdot z &= x \cdot z + y \cdot z \\ (-x) + x &= 0 & z \cdot (x + y) &= z \cdot x + z \cdot y \\ x + y &= y + x & x \cdot y &= y \cdot x \end{aligned}$$

Example 1.1.4. The “empty” theory with no operations and no equations is the theory of a set.

Example 1.1.5. The theory with one constant and no equations is the theory of a *pointed set*, cf. Example A.4.11.

Example 1.1.6. Let R be a ring. There is an algebraic theory of left R-modules. It has one constant 0, a unary operation $-$, a binary operation $+$, and for each $a \in R$ a unary operation $\overline{a}$, called *scalar multiplication by a*. The following equations hold:

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

For every $a, b \in R$ we also have the equations

$$\overline{a}(x+y) = \overline{a}\,x + \overline{a}\,y \;, \qquad \overline{a}(\overline{b}\,x) = \overline{(ab)}\,x \;, \qquad \overline{(a+b)}\,x = \overline{a}\,x + \overline{b}\,x \;.$$

Scalar multiplication by a is usually written as $a \cdot x$ instead of $\overline{a}\,x$. If we replace the ring R by a field $\mathbb{F}$ we obtain the algebraic theory of a vector space over $\mathbb{F}$ (even though the theory of fields is not algebraic!).

Example 1.1.7. In computer science, inductive datatypes are examples of algebraic theories. For example, the datatype of binary trees with leaves labeled by integers might be defined as follows in a programming language:

```
type tree = Leaf of int | Node of tree * tree
```

This corresponds to the algebraic theory with a constant `Leaf` n for each integer n and a binary operation `Node`. There are no equations. Actually, when computer scientists define a datatype like this, they have in mind a particular model of the theory, namely the *free* one.

Example 1.1.8. An obvious non-example is the theory of posets, formulated with a binary relation symbol $x \leq y$ and the usual axioms of reflexivity, transitivity and anti-symmetry, namely:

$$\begin{gathered} x \leq x \\ x \leq y \;,\; y \leq z \Rightarrow x \leq z \\ x \leq y \;,\; y \leq x \Rightarrow x = x \end{gathered}$$

On the other hand, using an operation of greatest lower bound or "meet" $x \wedge y$, one can make the equational theory of "$\wedge$-semilattices":

$$\begin{gathered} x \wedge x = x \\ x \wedge y = y \wedge x \\ x \wedge (y \wedge z) = (x \wedge y) \wedge z \end{gathered}$$

Then, defining a partial ordering by $x \leq y \iff x \wedge y = x$ we arrive at the notion of a "poset with meets", which *is* equational (of course, the same can be done with joins $x \vee y$ as well). We will show later (in section **??**) that there is no reformulation of the general theory of posets into an equivalent equational one.

Exercise 1.1.9. Let G be a group. Formulate the notion of a (left) G-set (i.e. a functor $G \to \mathsf{Set}$) as an algebraic theory.

1.1.1 Models of algebraic theories

Let us now consider *models* of an algebraic theory, i.e. *algebras*. Classically, a group can be given by a set G, an element $e \in G$, a function $m : G \times G \to G$ and a function $i : G \to G$, satisfying the group axioms:

$$\begin{aligned} m\,(x, m\,(y, z)) &= m\,(m\,(x, y)\,, z) \\ m\,(x, i\,x) &= m\,(i\,x, x) = e \\ m\,(x, e) &= m\,(e, x) = x \end{aligned}$$

for any $x, y, z \in G$. Observe, however, that this notion can easily be generalized so that we can speak of models of group theory in categories other than Set. This is accomplished simply by translating the equations between arbitrary elements into equations between the operations themselves: thus a group is given, first, by an object $G \in \mathsf{Set}$ and three morphisms

$$e : 1 \to G\,, \qquad m : G \times G \to G\,, \qquad i : G \to G\,.$$

The associativity axiom is then expressed by the commutativity of the following diagram:

$$\begin{tikzcd} G \times G \times G \arrow[r, "m \times \pi_2"] \arrow[d, "\pi_0 \times m"'] & G \times G \arrow[d, "m"] \\ G \times G \arrow[r, "m"'] & G \end{tikzcd} \tag{1.1}$$

Note that we have omitted the canonical associativity function $G \times (G \times G) \cong (G \times G) \times G$, which should be inserted into the top left corner of the diagram. The equations for the unit and the inverse are similarly expressed by commutativity of the following diagrams:

$$\begin{tikzcd} G \times 1 \arrow[r, "1_G \times e"] \arrow[dr, "\pi_0"'] & G \times G \arrow[d, "m"] & 1 \times G \arrow[l, "e \times 1_G"'] \arrow[dl, "\pi_1"] & G \arrow[r, "\langle 1_G, i \rangle"] \arrow[d, "!_G"'] & G \times G \arrow[d, "m"] & G \arrow[l, "\langle i, 1_G \rangle"'] \arrow[d, "!_G"] \\ & G & & 1 \arrow[r, "e"'] & G & 1 \arrow[l, "e"] \end{tikzcd} \tag{1.2}$$

This formulation makes sense in any category $\mathcal{C}$ with finite products.

Definition 1.1.10. Let $\mathcal{C}$ be a category with finite products. A *group in* $\mathcal{C}$ consists of an object G equipped with arrows:

$$\begin{tikzcd}
G \times G \arrow[r, "m"] & G & G \arrow[l, "i"'] \\
& 1 \arrow[u, "e"'] &
\end{tikzcd}$$

such that the above diagrams (1.1) and (1.2) expressing the group equations commute.

There is also an obvious corresponding generalization of a group homomorphism in Set to homomorphisms of groups in $\mathcal{C}$. Namely, an arrow in $\mathcal{C}$ between (the underlying objects of) groups, say $h : M \to N$, is a homomorphism if it commutes with the interpretations of the basic operations m, i, and e,

$$h \circ m^M = m^N \circ h^2 \qquad h \circ i^M = i^N \circ h \qquad h \circ e^M = e^N$$

as indicated in:

$$\begin{tikzcd}
M^2 \arrow[r, "h^2"] \arrow[d, "m^M"'] & N^2 \arrow[d, "m^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}
\qquad
\begin{tikzcd}
M \arrow[r, "h"] \arrow[d, "i^M"'] & N \arrow[d, "i^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "="] \arrow[d, "e^M"'] & 1 \arrow[d, "e^N"] \\
M \arrow[r, "h"'] & N
\end{tikzcd}$$

Together with the evident composition and identity arrows inherited from $\mathcal{C}$, this gives a category of groups in $\mathcal{C}$, which we denote:

$$\mathsf{Group}(\mathcal{C})$$

In general, we define an *interpretation* I of a theory $\mathbb{T}$ in a category $\mathcal{C}$ with finite products to consist of an object $I \in \mathcal{C}$ and, for each basic operation f of arity k, a morphism $f^I : I^k \to I$. (More formally, I is the tuple consisting of an underlying object $|I|$ and the interpretations f^I, but we shall write simply I for $|I|$.) In particular, basic constants are interpreted as morphisms $1 \to I$. The interpretation is then extended to all terms as follows: a general term t will be interpreted together with a *context of variables* $x_1, \ldots, x_n$ (a list without repetitions), where the variables appearing in t are among those appearing in the context. We write

$$x_1, \ldots, x_n \mid t \tag{1.3}$$

for a term t in context $x_1, \ldots x_n$. The interpretation of such a term in context (1.3) is a morphism $t^I : I^n \to I$, determined by the following specification:

1. The interpretation of a variable x_i among the $x_1, \ldots x_n$ is the i-th projection $\pi_i : I^n \to I$.

2. A term of the form $f\,(t_1, \ldots, t_k)$ is interpreted as the composite:

$$\begin{tikzcd} I^n \arrow[r, "({t_1}^I, \ldots, {t_k}^I)"] & I^k \arrow[r, "f^I"] & I \end{tikzcd}$$

   where ${t_i}^I : I^n \to I$ is the interpretation of the subterm t_i, for $i = 1, \ldots, k$, and f^I is the interpretation of the basic operation f.

It is clear that the interpretation of a term really depends on the context, and when necessary we shall write $t^I = [\,x_1, \ldots, x_n \mid t\,]^I$. For example, the term $f\,x_1$ is interpreted as a morphism $f^I : I \to I$ in context x_1, and as the morphism $f^I \circ \pi_1 : I^2 \to I$ in the context x_1, x_2.

Suppose u and v are terms in context $x_1, \ldots, x_n$. Then we say that the equation in context $x_1, \ldots, x_n \mid u = v$ is *satisfied* by the interpretation I if u^I and v^I are the same morphism in $\mathcal{C}$. In particular, if $u = v$ is an axiom of the theory, and $x_1, \ldots, x_n$ are all the variables appearing in either u or v, we say that I *satisfies the axiom* $u = v$, written

$$I \models u = v,$$

if $[x_1, \ldots, x_n \mid u]^I$ and $[x_1, \ldots, x_n \mid v]^I$ are the same morphism,

$$\begin{tikzcd} I^n \arrow[r, shift left, "{[x_1, \ldots, x_n \mid u]^I}"] \arrow[r, shift right, "{[x_1, \ldots, x_n \mid v]^I}"'] & I \end{tikzcd} \tag{1.4}$$

We can then define, as expected:

Definition 1.1.11 (cf. Definition **??**). A *model* M of an algebraic theory $\mathbb{T}$ in a category $\mathcal{C}$ with finite products (also called a $\mathbb{T}$-*algebra*) is an interpretation of the signature $\Sigma_{\mathbb{T}}$,

$$f^I : I^k \longrightarrow I$$

in $\mathcal{C}$, for all $f \in \Sigma_{\mathbb{T}}$, that satisfies the axioms $A_{\mathbb{T}}$,

$$I \models u = v,$$

for all $(u = v) \in A_{\mathbb{T}}$.

A *homomorphism* of models $h : M \to N$ is an arrow in $\mathcal{C}$ that commutes with the interpretations of the basic operations,

$$h \circ f^M = f^N \circ h^k$$

for all $f \in \Sigma_{\mathbb{T}}$, as indicated in:

$$\begin{tikzcd} M^k \arrow[r, "h^k"] \arrow[d, "f^M"'] & N^k \arrow[d, "f^N"] \\ M \arrow[r, "h"'] & N \end{tikzcd}$$

The category of $\mathbb{T}$-models in $\mathcal{C}$ is written,

$$\mathsf{Mod}(\mathbb{T}, \mathcal{C}).$$

A model of the empty theory $\mathbb{T}_0$ in $\mathcal{C}$ is therefore just an object A, and a homomorphism is just a map, so

$$\mathsf{Mod}(\mathbb{T}_0, \mathcal{C}) = \mathcal{C}.$$

A model of the theory $\mathbb{T}_{\mathsf{Group}}$ of groups in $\mathcal{C}$ is a group in $\mathcal{C}$, in the above sense, and similarly for homomorphisms, so:

$$\mathsf{Mod}(\mathbb{T}_{\mathsf{Group}}, \mathcal{C}) = \mathsf{Group}(\mathcal{C})$$

as defined above. In particular, a model in Set is just a group in the usual sense, so we have:

$$\mathsf{Mod}(\mathbb{T}_{\mathsf{Group}}, \mathsf{Set}) = \mathsf{Group}(\mathsf{Set}) = \mathsf{Group}.$$

An example of a new kind is provided the following.

Example 1.1.12. A model of the theory of groups in a functor category $\mathsf{Set}^{\mathbb{C}}$ is the same thing as a functor from $\mathbb{C}$ into the category groups,

$$\mathsf{Group}(\mathsf{Set}^{\mathbb{C}}) \cong \mathsf{Group}^{\mathbb{C}}.$$

Indeed, for each object $C \in \mathbb{C}$ there is an evaluation functor,

$$\mathsf{eval}_C : \mathsf{Set}^{\mathbb{C}} \to \mathsf{Set}$$

with $\mathsf{eval}_C(F) = F(C)$, and evaluation preserves products since these are computed pointwise in the functor category. Moreover, every arrow $h : C \to D$ in $\mathbb{C}$ gives rise to an obvious natural transformation $h : \mathsf{eval}_C \to \mathsf{eval}_D$. Thus for any group G in $\mathsf{Set}^{\mathbb{C}}$, we have groups $\mathsf{eval}_C(G)$ for each $C \in \mathbb{C}$ and group homomorphisms $h_G : C(G) \to D(G)$, comprising a functor $G : \mathbb{C} \to \mathsf{Group}$. Conversely, it is clear that a functor $H : \mathbb{C} \to \mathsf{Group}$ determines a group H in $\mathsf{Set}^{\mathbb{C}}$ with underlying object $|HC|$, where $| - | : \mathsf{Group} \to \mathsf{Set}$ is the forgetful functor. These constructions are clearly mutually inverse (up to canonical isomorphisms determined by the choice of products). In this way, *a group in the category of variable sets may be regarded as a variable group.*

Exercise 1.1.13. Verify the details of the isomorphism of categories

$$\mathsf{Mod}(\mathbb{T}, \mathsf{Set}^{\mathbb{C}}) \cong \mathsf{Mod}(\mathbb{T}, \mathsf{Set})^{\mathbb{C}}$$

discussed in example 1.1.12 for an arbitrary algebraic theory $\mathbb{T}$.

Exercise 1.1.14. Determine what a group is in the following categories: the category of graphs Graph, the category of topological spaces Top, and the category of groups Group. (Hint: Only the last case is tricky. Before thinking too hard about it, prove the following lemma [Bor94, Lemma 3.11.6], known as the Eckmann-Hilton argument. Let G be a set provided with two binary operations $\cdot$ and $\star$ and a common unit e, so that $x \cdot e = e \cdot x = x \star e = e \star x = x$. Suppose the two operations commute, i.e., $(x \star y) \cdot (z \star w) = (x \cdot z) \star (y \cdot w)$. Then they coincide, are *commutative* and associative.)

1.1.2 Theories as categories

The syntactically presented notion of an algebraic theory is a practical convenience, but as a specification of a mathematical concept, say that of a group, it has some defects. We would prefer a *presentation-free* notion that captures the group concept without tying it to a specific syntactic presentation (the example below indicates why). One such notion can be given by a category with a certain universal property, which determines it uniquely, up to equivalence of categories.

Let us consider group theory again. The algebraic axiomatization in terms of unit, multiplication and inverse is not the only possible one. For example, an alternative formulation uses the unit e and a binary operation $\odot$, called *double division*, along with a single axiom [McC93]:

$$(x \odot (((x \odot y) \odot z) \odot (y \odot e))) \odot (e \odot e) = z \;.$$

The usual group operations are related to double division as follows:

$$x \odot y = x^{-1} \cdot y^{-1} \;, \qquad x^{-1} = x \odot e \;, \qquad x \cdot y = (x \odot e) \odot (y \odot e) \;.$$

There may be practical reasons for prefering one formulation of group theory over another, but this should not determine what the general concept of a group is. For example, we would like to avoid particular choices of basic constants, operations, and axioms. This is akin to the situation where an algebra is presented by generators and relations: the algebra itself is regarded as independent of any particular such presentation. Similarly, one usually prefers a basis-free theory of vector spaces: it is better to formulate the general idea of a vector space without refering explicitly to a basis, even though every vector space has one.

As a first step, one could simply take *all* operations built from unit, multiplication, and inverse as basic, and *all* valid equations of group theory as axioms. But we can go a step further and collect all the operations into a category, thus forgetting about which ones were "basic" and which ones "derived", and which equalities were "axioms". We first describe this construction of a category $\mathcal{C}_{\mathbb{T}}$ for a general algebraic theory $\mathbb{T}$, and then determine another characterization of it.

As objects of $\mathcal{C}_{\mathbb{T}}$ we take *contexts*, i.e. sequences of distinct variables,

$$[x_1, \ldots, x_n] \;. \qquad (n \geq 0)$$

Actually, it will be convenient to take equivalence classes under renaming of variables, so that $[x_1, x_3] = [x_2, x_1]$. That is to say, the objects are just natural numbers.

A morphism from $[x_1, \ldots, x_m]$ to $[x_1, \ldots, x_n]$ is then an n-tuple $(t_1, \ldots, t_n)$, where each t_k is a term in the context $x_1, \ldots, x_m$, possibly after renaming the variables. Two such morphisms $(t_1, \ldots, t_n)$ and $(s_1, \ldots, s_n)$ are equal if, and only if, the axioms of the theory imply that $t_k = s_k$ for every $k = 1, \ldots, n$,

$$\mathbb{T} \vdash t_k = s_k$$

Strictly speaking, morphisms are thus (tuples of) *equivalence classes* of terms in context

$$[x_1, \ldots, x_m \mid t_1, \ldots, t_n\,] : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n],$$

where two terms are equivalent when the theory proves them to be equal (after renaming the variables). Since it is rather cumbersome to work with such equivalence classes, we shall work with the terms directly, but keeping in mind that equality between them is this equivalence. Note also that the context of the morphism agrees with its domain, so we can omit it from the notation when that domain is clear. The composition of morphisms

$$\begin{aligned}(t_1, \dots, t_m) &: [x_1, \dots, x_k] \to [x_1, \dots, x_m]\\ (s_1, \dots, s_n) &: [x_1, \dots, x_m] \to [x_1, \dots, x_n]\end{aligned}$$

is the morphism $(r_1, \dots, r_n)$ whose i-th component is obtained by simultaneously substituting in s_i the terms t_1, …, t_m for the variables x_1, …, x_m:

$$r_i = s_i[t_1, \dots, t_m/x_1, \dots, x_m] \qquad (1 \le i \le n)$$

The identity morphism on $[x_1, \dots, x_n]$ is $(x_1, \dots, x_n)$. Using the usual rules of deduction for equational logic (see Section B.5), it is easy to verify that these specifications are well-defined on equivalence classes, and therefore make $\mathcal{C}_\mathbb{T}$ a category.

Definition 1.1.15. The category $\mathcal{C}_\mathbb{T}$ just defined is called the *syntactic category* of the theory $\mathbb{T}$.

The syntactic category $\mathcal{C}_\mathbb{T}$ (which may be thought of as the "Lindenbaum-Tarski category" of $\mathbb{T}$, see **??**) contains the same "algebraic" information as the theory $\mathbb{T}$ from which it was built, but in a syntax-invariant way. Two different syntactic presentations of $\mathbb{T}$ — like the ones for groups mentioned above — will give rise to essentially the same category $\mathcal{C}_\mathbb{T}$ (i.e. up to isomorphism). In this sense, the category $\mathcal{C}_\mathbb{T}$ is the abstract, algebraic object presented by the "generators and relations" (the operations and equations) of $\mathbb{T}$. But there is another, still more important, sense in which $\mathcal{C}_\mathbb{T}$ represents $\mathbb{T}$, as we next show.

Exercise 1.1.16. Show that the syntactic category $\mathcal{C}_\mathbb{T}$ has all finite products.

1.1.3 Models as functors

Having represented an algebraic theory $\mathbb{T}$ by the syntactic category $\mathcal{C}_\mathbb{T}$ constructed from it, we next show that $\mathcal{C}_\mathbb{T}$ has the universal property that models of $\mathbb{T}$ correspond uniquely to certain functors from $\mathcal{C}_\mathbb{T}$. More precisely, given any category with finite products $\mathcal{C}$ (which we shall call an *FP-category*), there is a natural (in $\mathcal{C}$) equivalence,

$$\frac{\mathcal{M} \in \mathsf{Mod}(\mathbb{T}, \mathcal{C})}{\overline{\quad M : \mathcal{C}_\mathbb{T} \to \mathcal{C} \quad}} \tag{1.5}$$

between models $\mathcal{M}$ of $\mathbb{T}$ in $\mathcal{C}$ and finite product preserving functors ("*FP-functors*") $M : \mathcal{C}_\mathbb{T} \to \mathcal{C}$. The equivalence is mediated by a "universal model" $\mathcal{U}$ in $\mathcal{C}_\mathbb{T}$, corresponding to the identity functor $1_{\mathcal{C}_\mathbb{T}} : \mathcal{C}_\mathbb{T} \to \mathcal{C}_\mathbb{T}$ in the above displayed correspondence, so that every model $\mathcal{M}$ arises as the functorial image $M(\mathcal{U}) \cong \mathcal{M}$ of $\mathcal{U}$ under an essentially unique FP-functor

$M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$. The possibility of such universal models is an advantage of the generalized notion of a model in a category other than Set.

To give the details of the correspondence (1.5), let $\mathbb{T}$ be an arbitrary algebraic theory and $\mathcal{C}_{\mathbb{T}}$ the syntactic category constructed from $\mathbb{T}$ as in Definition 1.1.15. It is easy to show that the product in $\mathcal{C}_{\mathbb{T}}$ of two objects $[x_1, \ldots, x_n]$ and $[x_1, \ldots, x_m]$ is the object $[x_1, \ldots, x_{n+m}]$, and that $\mathcal{C}_{\mathbb{T}}$ has all finite products (including $\mathsf{1} = [-]$, the empty context). Moreover, there is a distinguished $\mathbb{T}$-model U in $\mathcal{C}_{\mathbb{T}}$ consisting of "the language itself", which we call the *syntactic model*: the underlying object is the context $U = [x_1]$ of length one, and each operation symbol f of, say, arity k is interpreted as itself,

$$f^U = [\, x_1, \ldots, x_k \mid f(x_1, \ldots, x_k)\,] : U^k = [x_1, \ldots, x_k] \longrightarrow [x_1] = U.$$

The axioms are all satisfied, because the equivalence relation on terms is determined by $\mathbb{T}$-provability. Explicitly, for terms s, t, we have:

$$U \models s = t \iff s^U = t^U \iff \mathbb{T} \vdash s = t. \tag{1.6}$$

We record this fact as the following.

Proposition 1.1.17. *The syntactic model U in $\mathcal{C}_{\mathbb{T}}$ is "generic" in the sense that it satisfies all and only the $\mathbb{T}$-provable equations.*

Even more importantly, though, the syntactic model U in $\mathcal{C}_{\mathbb{T}}$ has the following *universal property*:

Proposition 1.1.18. *Any model M in any finite product category $\mathcal{C}$ is the image of U under an essentially unique, finite product preserving functor $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$.*

In this sense, the syntactic category $\mathcal{C}_{\mathbb{T}}$ is the "free finite product category with a model of $\mathbb{T}$". The precise meaning of the proposition is given by the following proof. First, observe that any FP-functor $F : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ takes the syntactic model U in $\mathcal{C}_{\mathbb{T}}$ (or indeed any model) to a model FU in $\mathcal{C}$, with interpretations

$$f^{FU} = Ff^U : FU^k \to FU \qquad \text{for each } f \in \Sigma_k.$$

Moreover, any natural transformation $\vartheta : F \to G$ between FP-functors determines a homomorphism of models $h = \vartheta_U : FU \to GU$. In more detail, suppose $f : U \times U \to U$ is a basic operation, then there is a commutative diagram,

$$\begin{tikzcd}
FU \times FU \arrow[r, "h \times h"] \arrow[d, "\cong"] \arrow[dd, bend right=60, "f^{FU}"'] & GU \times GU \arrow[d, "\cong"'] \arrow[dd, bend left=60, "f^{GU}"] \\
F(U \times U) \arrow[r, "\vartheta_{U \times U}"] \arrow[d, "Ff"] & G(U \times U) \arrow[d, "Gf"'] \\
FU \arrow[r, "h = \vartheta_U"'] & GU
\end{tikzcd}$$

where the upper square commutes by preservation of products, and the lower one by naturality. Thus the operation "evaluation at U" determines a functor,

$$\mathsf{eval}_U : \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.7}$$

from the category of finite product preserving functors $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, with natural transformations as arrows, into the category of $\mathbb{T}$-models in $\mathcal{C}$.

Proposition 1.1.19. *The functor* (1.7) *is an equivalence of categories, natural in* $\mathcal{C}$*.*

Proof. Let M be any model in an FP-category $\mathcal{C}$. Then the assignment $f \mapsto f^M$ given by the interpretation part of M determines a functor $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, defined on objects by

$$M^\sharp[x_1, \ldots, x_k] = M^k$$

and on morphisms by

$$M^\sharp(t_1, \ldots, t_n) = \langle {t_1}^M, \ldots, {t_n}^M \rangle.$$

In more detail, $M^\sharp$ is defined on morphisms

$$[x_1, \ldots, x_k \mid t\,] : [x_1, \ldots, x_k] \to [x_1, \ldots, x_n]$$

in $\mathcal{C}_{\mathbb{T}}$ by the following rules:

1. The morphism
$$(x_i) : [x_1, \ldots, x_k] \to [x_1]$$
is mapped to the i-th projection
$$\pi_i : M^k \to M.$$

2. The morphism
$$(f\,(t_1,\ldots,t_m)) : [x_1,\ldots,x_k] \to [x_1]$$
is mapped to the composite
$$\begin{tikzcd} M^k \arrow[r, "(M^\sharp t_1, \ldots, M^\sharp t_m)"] & M^m \arrow[r, "M^\sharp f"] & M \end{tikzcd}$$
where $M^\sharp t_i : M^k \to M$ is the value of $M^\sharp$ on the morphisms $(t_i) : [x_1,\ldots,x_k] \to [x_1]$, for $i = 1,\ldots,m$, and $M^\sharp f = f^M$ is the interpretation of the basic operation f.

3. The morphism
$$(t_1,\ldots,t_n) : [x_1,\ldots,x_k] \to [x_1,\ldots,x_n]$$
is mapped to the morphism $\langle M^\sharp t_1,\ldots,M^\sharp t_n\rangle$ where $M^\sharp t_i$ is the value of $M^\sharp$ on the morphism $(t_i) : [x_1,\ldots,x_k] \to [x_1]$, and
$$\langle M^\sharp t_1,\ldots,M^\sharp t_n\rangle : M^k \longrightarrow M^n$$
is the evident n-tuple in the FP-category $\mathcal{C}$.

That $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ really is a functor now follows from the assumption that the interpretation M is a model, meaning that all the equations of the theory are satisfied by it, so that the above specification is well-defined on equivalence classes. Observe that the functor $M^\sharp$ is defined in such a way that it obviously preserves finite products, and that, moreover, there is an isomorphism of models,
$$M^\sharp(U) \cong M.$$

Thus we have shown that the functor "evaluation at U",
$$\mathsf{eval}_U : \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.8}$$
is essentially surjective on objects, since $\mathsf{eval}_U(M^\sharp) = M^\sharp(U) \cong M$.

We leave the verification that it is full and faithful as an easy exercise.

Exercise 1.1.20. Verify this.

Finally, naturality in $\mathcal{C}$ means the following. Suppose M is a model of $\mathbb{T}$ in any FP-category $\mathcal{C}$. Any FP-functor $F : \mathcal{C} \to \mathcal{D}$ to another FP-category $\mathcal{D}$ then takes M to a model $F(M)$ in $\mathcal{D}$. As for the special case of U, the interpretation is given by setting $f^{F(M)} = F(f^M)$ for the basic operations f (and composing with the canonical isos coming from preservation of products, $F(M) \times F(M) \cong F(M \times M)$, etc.). Since equations are described by commuting diagrams, F takes a model to a model, and the same is true for homomorphisms. Thus $F : \mathcal{C} \to \mathcal{D}$ induces a functor on $\mathbb{T}$-models,
$$\mathsf{Mod}(\mathbb{T}, F) : \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{D}).$$

By naturality of (1.7), we mean that the following square commutes up to natural isomorphism:

$$\begin{tikzcd}
\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \arrow[r, "\mathsf{eval}_U"] \arrow[d, "\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \arrow[d, "\mathsf{Mod}(\mathbb{T}, F)"] \\
\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{D}) \arrow[r, "\mathsf{eval}_U"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{C})
\end{tikzcd} \tag{1.9}$$

But this is clear, since for any FP-functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ we have:

$$\begin{aligned}
\mathsf{eval}_U \circ \mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)(M) &= (\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, F)(M))(U) \\
&= (F \circ M)(U) \\
&= F(M(U)) \\
&= F(\mathsf{eval}_U(M)) \\
&\cong \mathsf{Mod}(\mathbb{T}, F)(\mathsf{eval}_U(M)) \\
&= \mathsf{Mod}(\mathbb{T}, F) \circ \mathsf{eval}_U(M).
\end{aligned}$$

□

The equivalence of categories

$$\mathsf{Hom}_{\mathrm{FP}}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.10}$$

actually determines $\mathcal{C}_{\mathbb{T}}$ and the universal model U uniquely, up to equivalence of categories and isomorphism of models. Indeed, to recover U, just put $\mathcal{C}_{\mathbb{T}}$ for $\mathcal{C}$ and the identity functor $1_{\mathcal{C}_{\mathbb{T}}}$ on the left, to get U in $\mathsf{Mod}(\mathbb{T}, \mathcal{C}_{\mathbb{T}})$ on the right! To see that $\mathcal{C}_{\mathbb{T}}$ itself is also determined, observe that (1.10) says that the functor $\mathsf{Mod}(\mathbb{T}, \mathcal{C})$ is representable, with representing object $\mathcal{C}_{\mathbb{T}}$, in an appropriate (i.e. bicategorical) sense. As usual, this fact can also be formulated in elementary terms as a universal mapping property of $\mathcal{C}_{\mathbb{T}}$, as follows:

Definition 1.1.21. The *classifying category* of an algebraic theory $\mathbb{T}$ is an FP-category $\mathcal{C}_{\mathbb{T}}$ with a distinguished model U, called the *universal model*, such that:

(i) for any model M in any FP-category $\mathcal{C}$, there is an FP-functor

$$M^{\sharp} : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$$

and an isomorphism of models $M \cong M^{\sharp}(U)$.

(ii) for any FP-functors $F, G : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ and model homomorphism $h : F(U) \to G(U)$, there is a unique natural transformation $\vartheta : F \to G$ with

$$\vartheta_U = h.$$

Observe that (i) says that the evaluation functor (1.7) is essentially surjective, and (ii) that it is full and faithful. The category $\mathcal{C}_{\mathbb{T}}$ is then determined, up to equivalence, by this universal mapping property. Specifically, if $(\mathcal{C}, U)$ and $(\mathcal{D}, V)$ are both classifying categories for the same theory, then there are classifying functors,

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "V^\sharp"] & \mathcal{D} \arrow[l, bend left, "U^\sharp"]
\end{tikzcd}$$

the composites of which are necessarily isomorphic to the respective identity functors, since e.g. $U^\sharp(V^\sharp(U)) \cong U^\sharp(V) \cong U$.

We have now shown not only that every algebraic theory has a classifying category, but also that the syntactic category is essentially determined by that distinguishing property. We record this as the following.

Theorem 1.1.22. *Every algebraic theory $\mathbb{T}$ has a classifying category $\mathcal{C}_{\mathbb{T}}$, which can be constructed as the syntactic category of $\mathbb{T}$, in the sense of Definition 1.1.15.*

Example 1.1.23. Let us see explicitly what the foregoing definitions give us in the case of the theory of groups $\mathbb{T}_{\mathsf{Group}}$. Let us write $\mathbb{G} = \mathcal{C}_{\mathbb{T}_{\mathsf{Group}}}$ for the classifying category, which consists of contexts $[x_1, \dots, x_n]$ as objects, and terms built from variables and the group operations (modulo renaming and the group laws) as arrows. A finite product preserving functor $M : \mathbb{G} \to \mathsf{Set}$ is then determined uniquely, up to natural isomorphism, by its action on the context $[x_1]$ and the terms representing the basic operations. If we set

$$\begin{aligned} G &= M[x_1]\,, & e &= M(\cdot \mid e)\,, \\ i &= M(x_1 \mid {x_1}^{-1})\,, & m &= M(x_1, x_2 \mid x_1 \cdot x_2)\,, \end{aligned}$$

then (G, e, i, m) is just a group, with unit e, inverse i, and multiplication m. That G satisfies the axioms for groups follows from the functoriality of M and preservation of finite products, which implies preservation of the corresponding commutative diagrams. Conversely, any group (G, e, i, m) determines a finite product preserving functor $M_G : \mathbb{G} \to \mathsf{Set}$, by setting $M_G[x_1] = G$, etc. Thus $\mathsf{Mod}_{\mathsf{Set}}(\mathbb{G})$ will indeed be equivalent to Group once we show that both categories have the same notion of morphisms. This is shown just as in the general case above.

Example 1.1.24. Recall from 1.1.12 that a group G in the functor category $\mathsf{Set}^{\mathbb{C}}$ is essentially the same thing as a functor $G : \mathbb{C} \to \mathsf{Group}$. From the point of view of algebras as functors, this amounts to the observation that product-preserving functors $\mathbb{G} \to \mathsf{Hom}(\mathbb{C}, \mathsf{Set})$ correspond (by exponential transposition) to functors $\mathbb{C} \to \mathsf{Hom}_{\mathsf{FP}}(\mathbb{G}, \mathsf{Set})$, where the latter Hom-set consists just of product-preserving functors. Indeed, the correspondence extends to natural transformations to give the previously observed equivalance of categories,

$$\mathsf{Group}(\mathsf{Set}^{\mathbb{C}}) \simeq (\mathsf{Group}(\mathsf{Set}))^{\mathbb{C}} \simeq \mathsf{Group}^{\mathbb{C}}.$$

1.1.4 Completeness

Consider an algebraic theory $\mathbb{T}$ and an equation $s = t$ between terms of the theory. If the equation can be proved from the axioms of the theory, $\mathbb{T} \vdash s = t$, then every model M of the theory satisfies the equation, $M \models s = t$; this is the *soundness* of the equational calculus with respect to models in categories. The converse statement is:

$$M \models s = t \text{ for all } M \quad \Rightarrow \quad \mathbb{T} \vdash s = t\,.$$

This is called *completeness*, and (together with soundness) it says that the calculus of equations suffices for proving all (and only) the ones that hold in the semantics. For functorial semantics, this holds in an especially strong way: by Proposition 1.1.17, we already know that the syntactic model $U \in \mathsf{Mod}_{\mathbb{C}}(\mathbb{T})$ is generic, in the sense that satisfaction by U is equivalent to provability in $\mathbb{T}$,

$$U \models s = t \iff \mathbb{T} \vdash s = t.$$

But since U is also universal, it follows immediately that we also have soundness and completeness:

Theorem 1.1.25 (Soundness and Completeness)**.** *For any equation $s = t$, we have $\mathbb{T} \vdash s = t$ if and only if every model M in every FP-category $\mathcal{C}$ satisfies $s = t$.*

Proof. If $\mathbb{T} \vdash s = t$, then by Proposition 1.1.17 (the syntactic construction of $\mathcal{C}_{\mathbb{T}}$) we have $U \models s = t$, meaning that $s^U = t^U$. But then for any model M in an FP-category $\mathcal{C}$, we obtain $M \models s = t$ by applying the classifying functor $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$, which preserves the interpretations of s and t,

$$M^\sharp(s^U) = s^{M^\sharp(U)} = s^M$$

and so from $s^U = t^U$ we can infer $s^M = t^M$.

Conversely, if $M \models s = t$ for every model M, then in particular $U \models s = t$, and so $\mathbb{T} \vdash s = t$, since U is generic. □

Classically, it is seldom the case that there exists a generic model; instead, we consider the class of all models in Set. Completeness with respect to such a restricted class of models is of course a stronger statement than completeness with respect to all models in all categories. Toward the classical result, we first consider completeness with respect to "variable models" in Set, i.e. in presheaf categories $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.

Proposition 1.1.26. *Let $\mathbb{T}$ be an algebraic theory. The Yoneda embedding*

$$\mathsf{y} : \mathcal{C}_{\mathbb{T}} \to \widehat{\mathcal{C}_{\mathbb{T}}}$$

is a generic model for $\mathbb{T}$.

Proof. The Yoneda embedding $\mathsf{y} : \mathcal{C}_{\mathbb{T}} \to \widehat{\mathcal{C}_{\mathbb{T}}}$ preserves limits, and in particular finite products, hence it determines a model $\mathsf{y}U$ in the category of presheaves $\widehat{\mathcal{C}_{\mathbb{T}}}$. Like all models, $\mathsf{y}U$ satisfies all the equations that hold in U, simply because y is an FP functor, but because it is also faithful, any equation that holds in $\mathsf{y}U$ must already hold in U. □

Example 1.1.27. We consider group theory one last time. As a presheaf on (the classifying category of) the theory of groups $\mathbb{G}$, the generic group $\mathsf{y}U$ satisfies every equation that is satisfied by all groups, and no others. Let us describe its underlying object explicitly as a variable set. The presheaf $\mathsf{y}U$ is represented by the context with one variable, $U = [x_1]$,

$$\mathsf{y}U = \mathsf{y}[x_1] = \mathbb{G}(-, [x_1]) \ .$$

The values of this functor are thus a family of sets, parametrized by the objects of $\mathbb{G}$; namely, for every $n \in \mathbb{N}$, we have the set

$$\mathsf{y}U_n = \mathbb{G}([x_1, \ldots, x_n], [x_1])$$

consisting of all terms in n variables, modulo the equations of group theory; but this is just the set of elements of the *free group* $F(n)$ on n generators! Thus we have

$$\mathsf{y}U_n = \mathbb{G}([x_1, \ldots, x_n], [x_1]) \cong |F(n)| = \mathsf{Set}(1, |F(n)|) \cong \mathsf{Group}(F(1), F(n)).$$

Moreover, the unit, inverse, and multiplication operations on $\mathsf{y}U$ agree at each stage $\mathsf{y}U_n$ with the operations on the free group $F(n)$ (as the reader should verify).

To summarize, the presheaf of groups $\mathsf{y}U$ on the theory $\mathbb{G}$ is naturally isomorphic to the presheaf of free groups $F(x_1, \ldots, x_n)$ on n-generators. We will see below in more detail why this is so.

Finally, we consider completeness with respect to Set-valued models $M : \mathcal{C}_{\mathbb{T}} \to \mathsf{Set}$, which of course correspond to classical groups. We need the following:

Lemma 1.1.28. *For any small category $\mathbb{C}$, there is a jointly faithful family $(E_i)_{i \in I}$ of FP-functors $E_i : \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}} \to \mathsf{Set}$, with I a set. That is, for any maps $f, g : A \to B$ in $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, if $E_i(f) = E_i(g)$ for all $i \in I$, then $f = g$.*

Proof. We can take the evaluation functors $\mathrm{ev}_C : \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}} \to \mathsf{Set}$ for all $C \in \mathbb{C}$. These are clearly jointly faithful, and they preserve all limits and colimits, which are constructed pointwise in presheaves. □

Proposition 1.1.29. *Suppose $\mathbb{T}$ is an algebraic theory. For every equation $s = t$ between terms of the theory $\mathbb{T}$,*

$$M \models s = t \quad \textit{for all models } M \textit{ in } \mathsf{Set} \iff \mathbb{T} \vdash s = t.$$

Thus the equational calculus of algebraic theories is complete with respect to Set*-valued semantics.*

Proof. Combine the foregoing lemma with the fact, from Proposition 1.1.26, that the Yondea embedding is a generic model. □

Exercise 1.1.30. We described the object part of the representable functor $\mathsf{y}U$ given by the universal group U as a functor $\mathbb{G}^{\mathsf{op}} \to \mathsf{Set}$, in terms of the free groups $F(n)$. What is the action of $\mathsf{y}U$ on morphisms in $\mathbb{G}$, in these terms? Also describe the group structure on $\mathsf{y}U$ in $\widehat{\mathbb{G}}$ explicitly.

Exercise 1.1.31. Let s be a term of group theory with variables $x_1, \dots, x_n$. On the one hand we can think of s as an element of the free group $F(n)$, and on the other we can consider the interpretation of s with respect to the universal group U, namely as a natural transformation $s^U : U^n \Longrightarrow U$. Suppose t is another term of group theory with variables $x_1, \dots, x_n$. Show that $s^U = t^U$ if, and only if, $s = t$ in the free group $F(n)$.

1.1.5 Functorial semantics

Let us summarize our treatment of algebraic theories thus far. We have reformulated certain traditional *logical* notions in terms of *categorical* ones. The traditional approach may be described as involving the four different parts:

Type theory
There is an underlying type theory consisting of types and terms. For algebraic theories there is only one type, which is not even explicitly mentioned. The terms are built from variables and basic operations.

Logic Algebraic theories have a very simple kind of logic that involves only equations between terms and equational reasoning, i.e. substitution of equals for equals.

Theory
A theory is given by basic terms and axioms. The terms are expressed in the type theory, and the axioms are expressed in the logic.

Semantics
The type theory and logic can be interpreted in categories of the appropriate kind. For algebraic theories we used categories with finite products. The interpretation is determined by induction on the structure of the types, terms, and formulas (the equations). It is *denotational* in the sense that the types and terms of the theory "denote" the objects and morphisms. An interpretation of a theory is a *model* if it satisfies all the axioms of the theory, where in the present case the notion of satisfaction just means that the arrows interpreting the terms occurring in the equations are actually equal.

The alternative approach of *functorial semantics* developed here may be summarized as follows:

Theories are categories
From a given theory we constructed a category which expresses essentially the same information, but is syntax-invariant, in the sense that it does not depend on a particular presentation by (basic) operations and axioms. The structure of the category reflects the underlying type theory and logic. For example, single sorted algebraic theories give rise to categories with finite products.

Models are functors

A model is a (structure-preserving) functor from the category representing the theory to a category with appropriate structure to interpret the logic. The requirement that all axioms of the theory must be satisfied by a model translates to the requirement that the model is a functor, and that it preserves the structure of the category representing the theory. For models of algebraic theories, we only required that they preserve finite products, which along with functoriality ensures that all valid equations of the theory are preserved, and the axioms are therefore satisfied.

Homomorphisms are natural transformations

We obtain the notion of a homomorphisms of models for free: since models are functors, homomorphisms between them are natural transformations. Homomorphisms between models of algebraic theories turned out to be the usual notion of homomorphisms that respect the algebraic structure.

Universal model

By admitting models in categories other than Set, functorial semantics admits *universal models*: a model U in the classifying category $\mathcal{C}_{\mathbb{T}}$, such that any model anywhere is a functorial image of U by an essentially unique, structure-preserving functor. Such a universal model is then "logically generic", in the sense that it has all and only those logical properties had by all models, since such properties are preserved by the functors in question.

Logical completeness

The construction of the classifying category from the logical syntax of the theory shows the *soundness and completeness* of the theory with respect to general categorical semantics. Completeness with respect to a restricted class of models (e.g. Set-valued ones) results from an embedding theorem for the classifying category.

1.2 Lawvere duality

Appendix A

Category Theory

A.1 Categories

Definition A.1.1. A *category* $\mathcal{C}$ consists of classes

$\mathcal{C}_0$ of objects A, B, C, …
$\mathcal{C}_1$ of morphisms f, g, h, …

such that:

- Each morphism f has uniquely determined *domain* $\mathsf{dom}\, f$ and *codomain* $\mathsf{cod}\, f$, which are objects. This is written:
$$f : \mathsf{dom}\, f \to \mathsf{cod}\, f$$
- For any morphisms $f : A \to B$ and $g : B \to C$ there exists a uniquely determined *composition* $g \circ f : A \to C$. Composition is associative:
$$h \circ (g \circ f) = (h \circ g) \circ f\ ,$$
where domains are codomains are as follows:
$$\begin{tikzcd} A \arrow[r, "f"] & B \arrow[r, "g"] & C \arrow[r, "h"] & D \end{tikzcd}$$
- For every object A there exists the *identity* morphism $\mathsf{1}_A : A \to A$ which is a unit for composition,
$$\mathsf{1}_A \circ f = f\ , \qquad\qquad g \circ \mathsf{1}_A = g\ ,$$
where $f : B \to A$ and $g : A \to C$.

Morphisms are also called *arrows* or *maps*. Note that morphisms do not actually have to be functions, and objects need not be sets or spaces of any sort. We often write $\mathcal{C}$ instead of $\mathcal{C}_0$.

Definition A.1.2. A category $\mathcal{C}$ is *small* when the objects $\mathcal{C}_0$ and the morphisms $\mathcal{C}_1$ are sets (as opposed to proper classes). A category is *locally small* when for all objects $A, B \in \mathcal{C}_0$ the class of morphisms with domain A and codomain B, written $\mathsf{Hom}(A, B)$ or $\mathcal{C}_0(A, B)$, is a set.

We normally restrict attention to locally small categories, so unless we specify otherwise all categories are taken to be locally small. Next we consider several examples of categories.

A.1.1 Examples

The empty category $\mathsf{0}$ The empty category has no objects and no arrows.

The unit category $\mathsf{1}$ The unit category, also called the terminal category, has one object $\star$ and one arrow $1_\star$:

$$\begin{tikzcd} \star \arrow[loop right, "1_\star"] \end{tikzcd}$$

Other finite categories There are other finite categories, for example the category with two objects and one (non-identity) arrow, and the category with two parallel arrows:

$$\begin{tikzcd} \star \arrow[r] & \bullet \end{tikzcd} \qquad\qquad \begin{tikzcd} \star \arrow[r, bend left] \arrow[r, bend right] & \bullet \end{tikzcd}$$

Groups as categories Every group $(G, \cdot)$, is a category with a single object $\star$ and each element of G as a morphism:

$$\begin{tikzcd} \star \arrow[loop above, "b"] \arrow[loop right, "a"] \arrow[loop below, "c"] \end{tikzcd} \qquad\qquad a, b, c, \ldots \in G$$

The composition of arrows is given by the group operation:

$$a \circ b = a \cdot b$$

The identity arrow is the group unit e. This is indeed a category because the group operation is associative and the group unit is the unit for the composition. In order to get a category, we do not actually need to know that every element in G has an inverse. It suffices to take a *monoid*, also known as *semigroup*, which is an algebraic structure with an associative operation and a unit.

We can turn things around and *define* a monoid to be a category with a single object. A group is then a category with a single object in which every arrow is an *isomorphism* (in the sense of definition A.1.5 below).

Posets as categories Recall that a *partially ordered set*, or *poset* $(P, \leq)$, is a set with a reflexive, transitive, and antisymmetric relation:

$$\begin{aligned} x &\leq x && \text{(reflexive)} \\ x \leq y \ \& \ y \leq z &\Rightarrow x \leq z && \text{(transitive)} \\ x \leq y \ \& \ y \leq x &\Rightarrow x = y && \text{(antisymmetric)} \end{aligned}$$

Each poset is a category whose objects are the elements of P, and there is a single arrow $p \to q$ between $p, q \in P$ if, and only if, $p \leq q$. Composition of $p \to q$ and $q \to r$ is the unique arrow $p \to r$, which exists by transitivity of $\leq$. The identity arrow on p is the unique arrow $p \to p$, which exists by reflexivity of $\leq$.

Antisymmetry tells us that any two isomorphic objects in P are equal.[1] We do not need antisymmetry in order to obtain a category, i.e., a *preorder* would suffice.

Again, we may *define* a preorder to be a category in which there is at most one arrow between any two objects. A poset is a skeletal preorder, i.e. one in which the only isomorphisms are the identity arrows. We allow for the possibility that a preorder or a poset is a proper class rather than a set.

A particularly important example of a poset category is the poset of open sets $\mathcal{O}X$ of a topological space X, ordered by inclusion.

Sets as categories Any set S is a category whose objects are the elements of S and whose only arrows are identity arrows. Such a category, in which the only arrows are the identity arrows, is called a *discrete category.*

A.1.2 Categories of structures

In general, structures like groups, topological spaces, posets, etc., determine categories in which the maps are structure-preserving functions, composition is composition of functions, and identity morphisms are identity functions:

- Group is the category whose objects are groups and whose morphisms are group homomorphisms.
- Top is the category whose objects are topological spaces and whose morphisms are continuous maps.
- Set is the category whose objects are sets and whose morphisms are functions.[2]
- Graph is the category of (directed) graphs an graph homomorphisms.
- Poset is the category of posets and monotone maps.

[1] A category in which isomorphic object are equal is a *skeletal* category.

[2] A function between sets A and B is a relation $f \subseteq A \times B$ such that for every $x \in A$ there exists a unique $y \in B$ for which $\langle x, y \rangle \in f$. A morphism in Set is a triple $\langle A, f, B \rangle$ such that $f \subseteq A \times B$ is a function.

Such categories of structures are generally *large*, but locally small. Note that it is not necessary to check the associative and unit laws for such categories of functions (why?), unlike the following example.

Exercise A.1.3. The *category of relations* Rel has as objects all sets $A, B, C, \ldots$ and as arrows $A \to B$ the relations $R \subseteq A \times B$. The composite of $R \subseteq A \times B$ and $S \subseteq B \times C$, and the identity arrow on A, are defined by:

$$\begin{aligned} S \circ R &= \big\{ \langle x, z \rangle \in A \times C \;\big|\; \exists\, y \in B \,.\, xRy \;\&\; ySz \big\} \,, \\ 1_A &= \big\{ \langle x, x \rangle \;\big|\; x \in A \big\} \,. \end{aligned}$$

Show that this is indeed a category!

A.1.3 Basic notions

We recall some further basic notions from category theory.

Definition A.1.4. A *subcategory* $\mathcal{C}'$ of a category $\mathcal{C}$ is given by a subclass of objects $\mathcal{C}'_0 \subseteq \mathcal{C}_0$ and a subclass of morphisms $\mathcal{C}'_1 \subseteq \mathcal{C}_1$ such that $f \in \mathcal{C}'_1$ implies $\mathsf{dom}\, f, \mathsf{cod}\, f \in \mathcal{C}'_0$, $1_A \in \mathcal{C}'_1$ for every $A \in \mathcal{C}'_0$, and $g \circ f \in \mathcal{C}'_1$ whenever $f, g \in \mathcal{C}'_1$ are composable.

A subcategory $\mathcal{C}'$ of $\mathcal{C}$ is *full* if for all $A, B \in \mathcal{C}'_0$, we have $\mathcal{C}'(A, B) = \mathcal{C}(A, B)$, i.e. every $f : A \to B$ in $\mathcal{C}_1$ is also in $\mathcal{C}'_1$.

Definition A.1.5. An *inverse* of a morphism $f : A \to B$ is a morphism $f^{-1} : B \to A$ such that

$$f \circ f^{-1} = 1_B \qquad\qquad \text{and} \qquad\qquad f^{-1} \circ f = 1_A \;.$$

A morphism that has an inverse is an *isomorphism*, or *iso*. If there exists a pair of mutually inverse morphisms $f : A \to B$ and $f^{-1} : B \to A$ we say that the objects A and B are *isomorphic*, written $A \cong B$.

The notation f^{-1} is justified because an inverse, if it exists, is unique. A *left inverse* is a morphism $g : B \to A$ such that $g \circ f = 1_A$, and a *right inverse* is a morphism $g : B \to A$ such that $f \circ g = 1_B$. A left inverse is also called a *retraction*, whereas a right inverse is called a *section.*

Definition A.1.6. A *monomorphism*, or *mono*, is a morphism $f : A \to B$ that can be cancelled on the left: for all $g : C \to A$, $h : C \to A$,

$$f \circ g = f \circ h \Rightarrow g = h \;.$$

An *epimorphism*, or *epi*, is a morphism $f : A \to B$ that can be cancelled on the right: for all $g : B \to C$, $h : B \to A$,

$$g \circ f = h \circ f \Rightarrow g = h \;.$$

In Set monomorphisms are the injective functions and epimorphisms are the surjective functions. Isomorphisms in Set are the bijective functions. Thus, in Set a morphism is iso if, and only if, it is both mono and epi. However, this example is misleading! In general, a morphism can be mono and epi without being an iso. For example, the non-identity morphism in the category consisting of two objects and one morphism between them is both epi and mono, but it has no inverse. A more interesting example of morphisms that are both epi and mono but are not iso occurs in the category Top of topological spaces and continuous maps, where not every continuous bijection is a homeomorphism.

A *diagram* of objects and morphisms is a directed graph whose vertices are objects of a category and edges are morphisms between them, for example:

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[r, "h"] & C \arrow[dl, "j"] \\
D \arrow[ur, "m"'] \arrow[r, "k"'] & E &
\end{tikzcd}$$

Such a diagram is said to *commute* when the composition of morphisms along any two paths with the same beginning and end gives equal morphisms. Commutativity of the above diagram is equivalent to the following two equations:

$$f = m \circ g \,, \qquad\qquad k = j \circ h \circ m \,.$$

From these we can derive $k \circ g = j \circ h \circ f$ by a *diagram chase*.

A.2 Functors

Definition A.2.1. A *functor* $F : \mathcal{C} \to \mathcal{D}$ from a category $\mathcal{C}$ to a category $\mathcal{D}$ consists of functions

$$F_0 : \mathcal{C}_0 \to \mathcal{D}_0 \qquad\qquad \text{and} \qquad\qquad F_1 : \mathcal{C}_1 \to \mathcal{D}_1$$

such that, for all $f : A \to B$ and $g : B \to C$ in $\mathcal{C}$:

$$\begin{aligned} F_1 f &: F_0 A \to F_0 B \,, \\ F_1(g \circ f) &= (F_1 g) \circ (F_1 f) \,, \\ F_1(1_A) &= 1_{F_0 A} \,. \end{aligned}$$

We usually write F for both F_0 and F_1.

A functor is thus a homomorphism of the category structure; note that it maps commutative diagrams to commutative diagrams because it preserves composition.

We may form the "category of categories" Cat whose objects are small categories and whose morphisms are functors. Composition of functors is composition of the corresponding functions, and the identity functor is one that is identity on objects and on morphisms. The category Cat is large but locally small.

Definition A.2.2. A functor $F : \mathcal{C} \to \mathcal{D}$ is *faithful* when it is "locally injective on morphisms", in the sense that for all $f, g : A \to B$, if $Ff = Fg$ then $f = g$.

A functor $F : \mathcal{C} \to \mathcal{D}$ is *full* when it is "locally surjective on morphisms": for every $g : FA \to FB$ there exists $f : A \to B$ such that $g = Ff$.

We consider several examples of functors.

A.2.1 Functors between sets, monoids and posets

When sets, monoids, groups, and posets are regarded as categories, the functors turn out to be the *usual morphisms*, for example:

- A functor between sets S and T is a function from S to T.
- A functor between groups G and H is a group homomorphism from G to H.
- A functor between posets P and Q is a monotone function from P to Q.

Exercise A.2.3. Verify that the above claims are correct.

A.2.2 Forgetful functors

For categories of structures Group, Top, Graph, Poset, …, there is a *forgetful* functor U which maps an object to the underlying set and a morphism to the underlying function. For example, the forgetful functor $U : \mathsf{Group} \to \mathsf{Set}$ maps a group $(G, \cdot)$ to the set G and a group homomorphism $f : (G, \cdot) \to (H, \star)$ to the function $f : G \to H$.

There are also forgetful functors that forget only part of the structure, for example the forgetful functor $U : \mathsf{Ring} \to \mathsf{Group}$ which maps a ring $(R, +, \times)$ to the additive group $(R, +)$ and a ring homomorphism $f : (R, +_R, \cdot_S) \to (S, +_S, \cdot_S)$ to the group homomorphism $f : (R, +_R) \to (S, +_S)$. Note that there is another forgetful functor $U' : \mathsf{Ring} \to \mathsf{Mon}$ from rings to monoids.

Exercise A.2.4. Show that taking the graph $\Gamma(f) = \{\langle x, f(x)\rangle \mid x \in A\}$ of a function $f : A \to B$ determines a functor $\Gamma : \mathsf{Set} \to \mathsf{Rel}$, from sets and functions to sets and relations, which is the identity on objects. Is this a forgetful functor?

A.3 Constructions of Categories and Functors

A.3.1 Product of categories

Given categories $\mathcal{C}$ and $\mathcal{D}$, we form the *product category* $\mathcal{C} \times \mathcal{D}$ whose objects are pairs of objects $\langle C, D\rangle$ with $C \in \mathcal{C}$ and $D \in \mathcal{D}$, and whose morphisms are pairs of morphisms $\langle f, g\rangle : \langle C, D\rangle \to \langle C', D'\rangle$ with $f : C \to C'$ in $\mathcal{C}$ and $g : D \to D'$ in $\mathcal{D}$. Composition is given by $\langle f, g\rangle \circ \langle f', g'\rangle = \langle f \circ f', g \circ g'\rangle$.

There are evident *projection* functors

$$\begin{tikzcd} & \mathcal{C} \times \mathcal{D} \arrow[dl, "\pi_0"'] \arrow[dr, "\pi_1"] & \\ \mathcal{C} & & \mathcal{D} \end{tikzcd}$$

which act as indicated in the following diagrams:

$$\begin{tikzcd} & \langle C, D \rangle \arrow[dl, mapsto, "\pi_0"'] \arrow[dr, mapsto, "\pi_1"] & & & \langle f, g \rangle \arrow[dl, mapsto, "\pi_0"'] \arrow[dr, mapsto, "\pi_1"] & \\ C & & D & f & & g \end{tikzcd}$$

Exercise A.3.1. Show that, for any categories $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, there are distinguished isos:

$$\mathsf{1} \times \mathbb{C} \cong \mathbb{C}$$
$$\mathbb{B} \times \mathbb{C} \cong \mathbb{C} \times \mathbb{B}$$
$$\mathbb{A} \times (\mathbb{B} \times \mathbb{C}) \cong (\mathbb{A} \times \mathbb{B}) \times \mathbb{C}$$

Does this make Cat a (commutative) monoid?

A.3.2 Slice categories

Given a category $\mathcal{C}$ and an object $A \in \mathcal{C}$, the *slice* category $\mathcal{C}/A$ has as objects, morphisms into A,

$$\begin{tikzcd} B \arrow[d, "f"] \\ A \end{tikzcd} \tag{A.1}$$

and as morphisms, commutative diagrams over A:

$$\begin{tikzcd} B \arrow[rr, "g"] \arrow[dr, "f"'] & & B' \arrow[dl, "f'"] \\ & A & \end{tikzcd} \tag{A.2}$$

That is, a morphism from $f : B \to A$ to $f' : B' \to A$ is a morphism $g : B \to B'$ such that $f = f' \circ g$. Composition of morphisms in $\mathcal{C}/A$ is composition of morphisms in $\mathcal{C}$.

There is a forgetful functor $U_A : \mathcal{C}/A \to \mathcal{C}$ which maps an object (A.1) to its domain B, and a morphism (A.2) to the morphism $g : B \to B'$.

Furthermore, for each morphism $h : A \to A'$ in $\mathcal{C}$ there is a functor "composition by h",

$$\mathcal{C}/h : \mathcal{C}/A \to \mathcal{C}/A'$$

which maps an object (A.1) to the object $h \circ f : B \to A'$ and a morphisms (A.2) to the morphism

$$\begin{tikzcd}
B \arrow[rr, "g"] \arrow[dr, "h \circ f"'] & & B' \arrow[dl, "h \circ f'"] \\
& A' &
\end{tikzcd}$$

The construction of slice categories is itself a functor

$$\mathcal{C}/- : \mathcal{C} \to \mathsf{Cat}$$

provided that $\mathcal{C}$ is small. This functor maps each $A \in \mathcal{C}$ to the category $\mathcal{C}/A$ and each morphism $h : A \to A'$ to the composition functor $\mathcal{C}/h : \mathcal{C}/A \to \mathcal{C}/A'$.

Since Cat is itself a category, we may form the slice category $\mathsf{Cat}/\mathcal{C}$ for any small category $\mathcal{C}$. The slice functor $\mathcal{C}/-$ then factors through the forgetful functor $U_{\mathcal{C}} : \mathsf{Cat}/\mathcal{C} \to \mathsf{Cat}$ via a functor $\overline{\mathcal{C}} : \mathcal{C} \to \mathsf{Cat}/\mathcal{C}$,

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "\overline{\mathcal{C}}"] \arrow[dr, "\mathcal{C}/-"'] & \mathsf{Cat}/\mathcal{C} \arrow[d, "U_{\mathcal{C}}"] \\
& \mathsf{Cat}
\end{tikzcd}$$

where for $A \in \mathcal{C}$, the object part $\overline{\mathcal{C}}A$ is

$$\begin{tikzcd}
\mathcal{C}/A \arrow[d, "U_A"] \\
\mathcal{C}
\end{tikzcd}$$

and for $h : A \to A'$ in $\mathcal{C}$, the morphism part $\overline{\mathcal{C}}h$ is

$$\begin{tikzcd}
\mathcal{C}/A \arrow[rr, "\mathcal{C}/h"] \arrow[dr, "U_A"'] & & \mathcal{C}/A' \arrow[dl, "U_{A'}"] \\
& \mathcal{C} &
\end{tikzcd}$$

A.3.3 Arrow categories

Similar to the slice categories, an arrow category has arrows as objects, but without a fixed codomain. Given a category $\mathcal{C}$, the *arrow* category $\mathcal{C}^{\to}$ has as objects the morphisms of $\mathcal{C}$,

$$\begin{tikzcd}
A \arrow[d, "f"] \\
B
\end{tikzcd} \tag{A.3}$$

and as morphisms $f \to f'$ the commutative squares,

$$\begin{tikzcd}
A \arrow[r, "g"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\
B \arrow[r, "g'"'] & B'.
\end{tikzcd} \tag{A.4}$$

That is, a morphism from $f : A \to B$ to $f' : A' \to B'$ is a pair of morphisms $g : A \to A'$ and $g' : B \to B'$ such that $g' \circ f = f' \circ g$. Composition of morphisms in $\mathcal{C}^{\to}$ is just componentwise composition of morphisms in $\mathcal{C}$.

There are two evident forgetful functors $U_1, U_2 : \mathcal{C}^{\to} \to \mathcal{C}$, given by the domain and codomain operations. (Can you find a common section for these?)

A.3.4 Opposite categories

For a category $\mathcal{C}$ the *opposite category* $\mathcal{C}^{\mathsf{op}}$ has the same objects as $\mathcal{C}$, but all the morphisms are turned around, that is, a morphism $f : A \to B$ in $\mathcal{C}^{\mathsf{op}}$ is a morphism $f : B \to A$ in $\mathcal{C}$. The identity arrows in $\mathcal{C}^{\mathsf{op}}$ are the same as in $\mathcal{C}$, but the order of composition is reversed. The opposite of the opposite of a category is clearly the original category.

A functor $F : \mathcal{C}^{\mathsf{op}} \to \mathcal{D}$ is sometimes called a *contravariant functor* (from $\mathcal{C}$ to $\mathcal{D}$), and a functor $F : \mathcal{C} \to \mathcal{D}$ is a *covariant* functor.

For example, the opposite category of a preorder $(P, \leq)$ is the preorder P turned upside down, $(P, \geq)$.

Exercise A.3.2. Given a functor $F : \mathcal{C} \to \mathcal{D}$, can you define a functor $F^{\mathsf{op}} : \mathcal{C}^{\mathsf{op}} \to \mathcal{D}^{\mathsf{op}}$ in such a way that $-^{\mathsf{op}}$ itself becomes a functor? On what category is it a functor?

A.3.5 Representable functors

Let $\mathcal{C}$ be a locally small category. Then for each pair of objects $A, B \in \mathcal{C}$ the collection of all morphisms $A \to B$ forms a set, written $\mathsf{Hom}_{\mathcal{C}}(A, B)$, $\mathsf{Hom}(A, B)$ or $\mathcal{C}(A, B)$. For every $A \in \mathcal{C}$ there is a functor

$$\mathcal{C}(A, -) : \mathcal{C} \to \mathsf{Set}$$

defined by

$$\begin{aligned}
&\mathcal{C}(A, B) = \big\{ f \in \mathcal{C}_1 \;\big|\; f : A \to B \big\} \\
&\mathcal{C}(A, g) : f \mapsto g \circ f
\end{aligned}$$

where $B \in \mathcal{C}$ and $g : B \to C$. In words, $\mathcal{C}(A, g)$ is composition by g. This is indeed a functor because, for any morphisms

$$\begin{tikzcd}
A \arrow[r, "f"] & B \arrow[r, "g"] & C \arrow[r, "h"] & D
\end{tikzcd} \tag{A.5}$$

we have

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

and $\mathcal{C}(A, 1_B)f = 1_A \circ f = f = 1_{\mathcal{C}(A,B)} f$.

We may also ask whether $\mathcal{C}(-, B)$ is a functor. If we define its action on morphisms to be precomposition,

$$\mathcal{C}(f, B) : g \mapsto g \circ f \ ,$$

it becomes a *contravariant* functor,

$$\mathcal{C}(-, B) : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set} \ .$$

The contravariance is a consequence of precomposition; for morphisms (A.5) we have

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

A functor of the form $\mathcal{C}(A, -)$ is a *(covariant) representable functor*, and a functor of the form $\mathcal{C}(-, B)$ is a *(contravariant) representable functor*.

It follows that the hom-set is a functor

$$\mathcal{C}(-, -) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathsf{Set}$$

which maps a pair of objects $A, B \in \mathcal{C}$ to the set $\mathcal{C}(A, B)$ of morphisms from A to B, and it maps a pair of morphisms $f : A' \to A$, $g : B \to B'$ in $\mathcal{C}$ to the function

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

defined by

$$\mathcal{C}(f, g) : h \mapsto g \circ h \circ f \ .$$

(Why does it follow that this is a functor?)

A.3.6 Group actions

A group $(G, \cdot)$ is a category with one object $\star$ and elements of G as the morphisms. Thus, a functor $F : G \to \mathsf{Set}$ is given by a set $F\star = S$ and for each $a \in G$ a function $Fa : S \to S$ such that, for all $x \in S$, $a, b \in G$,

$$(Fe)x = x \ , \qquad\qquad (F(a \cdot b))x = (Fa)((Fb)x) \ .$$

Here e is the unit element of G. If we write $a \cdot x$ instead of $(Fa)x$, the above two equations become the familiar laws for a *left group action on the set S*:

$$e \cdot x = x \ , \qquad\qquad (a \cdot b) \cdot x = a \cdot (b \cdot x) \ .$$

Exercise A.3.3. A *right group action* by a group $(G, \cdot)$ on a set S is an operation $\cdot : S \times G \to S$ that satisfies, for all $x \in S$, $a, b \in G$,

$$x \cdot e = x \ , \qquad\qquad x \cdot (a \cdot b) = (x \cdot a) \cdot b \ .$$

Exhibit right group actions as functors.

A.4 Natural Transformations and Functor Categories

Definition A.4.1. Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C} \to \mathcal{D}$ be functors. A *natural transformation* $\eta : F \Longrightarrow G$ from F to G is a map $\eta : \mathcal{C}_0 \to \mathcal{D}_1$ which assigns to every object $A \in \mathcal{C}$ a morphism $\eta_A : FA \to GA$, called the *component of* η *at* A, such that for every $f : A \to B$ in $\mathcal{C}$ we have $\eta_B \circ Ff = Gf \circ \eta_A$, i.e., the following diagram in $\mathcal{D}$ commutes:

$$\begin{tikzcd}
FA \arrow[r, "\eta_A"] \arrow[d, "Ff"'] & GA \arrow[d, "Gf"] \\
FB \arrow[r, "\eta_B"'] & GB
\end{tikzcd}$$

A simple example is given by the “twist” isomorphism $t : A \times B \to B \times A$ (in Set). Given any maps $f : A \to A'$ and $g : B \to B'$, there is a commutative square:

$$\begin{tikzcd}
A \times B \arrow[r, "t_{A,B}"] \arrow[d, "f \times g"'] & B \times A \arrow[d, "g \times f"] \\
A' \times B' \arrow[r, "t_{A',B'}"'] & B' \times A'
\end{tikzcd}$$

Thus naturality means that the two *functors* $F(X, Y) = X \times Y$ and $G(X, Y) = Y \times X$ are related to each other (by $t : F \to G$), and not simply their individual values $A \times B$ and $B \times A$. As a further example of a natural transformation, consider groups G and H as categories and two homomorphisms $f, g : G \to H$ as functors between them. A natural transformation $\eta : f \Longrightarrow g$ is given by a single element $\eta_\star = b \in H$ such that, for every $a \in G$, the following diagram commutes:

$$\begin{tikzcd}
\star \arrow[r, "b"] \arrow[d, "fa"'] & \star \arrow[d, "ga"] \\
\star \arrow[r, "b"'] & \star
\end{tikzcd}$$

This means that $b \cdot fa = (ga) \cdot b$, that is $ga = b \cdot (fa) \cdot b^{-1}$. In other words, a natural transformation $f \Longrightarrow g$ is a *conjugation* operation $b^{-1} \cdot - \cdot b$ which transforms f into g.

For every functor $F : \mathcal{C} \to \mathcal{D}$ there exists the *identity transformation* $1_F : F \Longrightarrow F$ defined by $(1_F)_A = 1_A$. If $\eta : F \Longrightarrow G$ and $\theta : G \Longrightarrow H$ are natural transformations, then their composition $\theta \circ \eta : F \Longrightarrow H$, defined by $(\theta \circ \eta)_A = \theta_A \circ \eta_A$ is also a natural transformation. Composition of natural transformations is associative because it is composition in the codomain category $\mathcal{D}$. This leads to the definition of functor categories.

Definition A.4.2. Let $\mathcal{C}$ and $\mathcal{D}$ be categories. The *functor category* $\mathcal{D}^{\mathcal{C}}$ is the category whose objects are functors from $\mathcal{C}$ to $\mathcal{D}$ and whose morphisms are natural transformations between them.

A functor category may be quite large, too large in fact. In order to avoid problems with size we normally require $\mathcal{C}$ to be a locally small category. The “hom-class” of all natural transformations $F \Longrightarrow G$ is usually written as

$$\mathsf{Nat}(F, G)$$

instead of the more awkward $\mathsf{Hom}_{\mathcal{D}^{\mathcal{C}}}(F, G)$.

Suppose we have functors F, G, and H with a natural transformation $\theta : G \Longrightarrow H$, as in the following diagram:

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "F"] & \mathcal{D} \arrow[r, bend left, "G"] \arrow[r, bend right, "H"'] \arrow[r, phantom, "\Downarrow\theta"] & \mathbb{E}
\end{tikzcd}$$

Then we can form a natural transformation $\theta \circ F : G \circ F \Longrightarrow H \circ F$ whose component at $A \in \mathcal{C}$ is $(\theta \circ F)_A = \theta_{FA}$.

Similarly, if we have functors and a natural transformation

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "G"] \arrow[r, bend right, "H"'] \arrow[r, phantom, "\Downarrow\theta"] & \mathcal{D} \arrow[r, "F"] & \mathbb{E}
\end{tikzcd}$$

we can form a natural transformation $(F \circ \theta) : F \circ G \Longrightarrow F \circ H$ whose component at $A \in \mathcal{C}$ is $(F \circ \theta)_A = F\theta_A$. These operations are known as *whiskering.*

A *natural isomorphism* is an isomorphism in a functor category. Thus, if $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C} \to \mathcal{D}$ are two functors, a natural isomorphism between them is a natural transformation $\eta : F \Longrightarrow G$ whose components are isomorphisms. In this case, the inverse natural transformation $\eta^{-1} : G \Longrightarrow F$ is given by $(\eta^{-1})_A = (\eta_A)^{-1}$. We write $F \cong G$ when F and G are naturally isomorphic.

The definition of natural transformations is motivated in part by the fact that, for any small categories $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, we have

$$\mathsf{Cat}(\mathbb{A} \times \mathbb{B}, \mathbb{C}) \cong \mathsf{Cat}(\mathbb{A}, \mathbb{C}^{\mathbb{B}}) \; . \tag{A.6}$$

The isomorphism takes a functor $F : \mathbb{A} \times \mathbb{B} \to \mathbb{C}$ to the functor $\widetilde{F} : \mathbb{A} \to \mathbb{C}^{\mathbb{B}}$ defined on objects $A \in \mathbb{A}$, $B \in \mathbb{B}$ by

$$(\widetilde{F}A)B = F\langle A, B\rangle$$

and on a morphism $f : A \to A'$ by

$$(\widetilde{F}f)_B = F\langle f, 1_B\rangle \; .$$

The functor $\widetilde{F}$ is called the *transpose* of F.

The inverse isomorphism takes a functor $G : \mathbb{A} \to \mathbb{C}^{\mathbb{B}}$ to the functor $\widetilde{G} : \mathbb{A} \times \mathbb{B} \to \mathbb{C}$, defined on objects by

$$\widetilde{G}\langle A, B\rangle = (GA)B$$

and on a morphism $\langle f, g\rangle : A \times B \to A' \times B'$ by

$$\widetilde{G}\langle f, g\rangle = (Gf)_{B'} \circ (GA)g = (GA')g \circ (Gf)_B \ ,$$

where the last equation holds by naturality of Gf:

$$\begin{tikzcd}
(GA)B \arrow[r, "(Gf)_B"] \arrow[d, "(GA)g"'] & (GA')B \arrow[d, "(GA')g"] \\
(GA)B' \arrow[r, "(Gf)_{B'}"'] & (GA')B'
\end{tikzcd}$$

A.4.1 Directed graphs as a functor category

Recall that a *directed graph* G is given by a set of vertices G_V and a set of edges G_E. Each edge $e \in G_E$ has a uniquely determined *source* $\mathsf{src}_G\, e \in G_V$ and *target* $\mathsf{trg}_G\, e \in G_V$. We write $e : a \to b$ when a is the source and b is the target of e. A *graph homomorphism* $\phi : G \to H$ is a pair of functions $\phi_0 : G_V \to H_V$ and $\phi_1 : G_E \to H_E$, where we usually write ϕ for both ϕ_0 and ϕ_1, such that whenever $e : a \to b$ then $\phi_1 e : \phi_0 a \to \phi_0 b$. The category of directed graphs and graph homomorphisms is denoted by Graph.

Now let $\cdot \rightrightarrows \cdot$ be the category with two objects and two parallel morphisms, depicted by the following "sketch":

$$\begin{tikzcd}
E \arrow[r, bend left, "s"] \arrow[r, bend right, "t"'] & V
\end{tikzcd}$$

An object of the functor category $\mathsf{Set}^{\cdot\rightrightarrows\cdot}$ is a functor $G : (\cdot \rightrightarrows \cdot) \to \mathsf{Set}$, which consists of two sets GE and GV and two functions $Gs : GE \to GV$ and $Gt : GE \to GV$. But this is precisely a directed graph whose vertices are GV, the edges are GE, the source of $e \in GE$ is $(Gs)e$ and the target is $(Gt)e$. Conversely, any directed graph G is a functor $G : (\cdot \rightrightarrows \cdot) \to \mathsf{Set}$, defined by

$$GE = G_E \ , \qquad GV = G_V \ , \qquad Gs = \mathsf{src}_G \ , \qquad Gt = \mathsf{trg}_G \ .$$

Now category theory begins to show its worth, for the morphisms in $\mathsf{Set}^{\cdot\rightrightarrows\cdot}$ are precisely the graph homomorphisms. Indeed, a natural transformation $\phi : G \Longrightarrow H$ between graphs is a pair of functions,

$$\phi_E : G_E \to H_E \qquad \text{and} \qquad \phi_V : G_V \to H_V$$

whose naturality is expressed by the commutativity of the following two diagrams:

$$\begin{tikzcd}
G_E \arrow[r, "\phi_E"] \arrow[d, "\mathsf{src}_G"'] & H_E \arrow[d, "\mathsf{src}_H"] \\
G_V \arrow[r, "\phi_V"'] & H_V
\end{tikzcd}
\qquad
\begin{tikzcd}
G_E \arrow[r, "\phi_E"] \arrow[d, "\mathsf{trg}_G"'] & H_E \arrow[d, "\mathsf{trg}_H"] \\
G_V \arrow[r, "\phi_V"'] & H_V
\end{tikzcd}$$

This is precisely the requirement that $e : a \to b$ implies $\phi_E e : \phi_V a \to \phi_V b$. Thus, in sum, we have,

$$\mathsf{Graph} = \mathsf{Set}^{\cdot \rightrightarrows \cdot}.$$

Exercise A.4.3. Exhibit the arrow category $\mathcal{C}^{\to}$ and the category of group actions $\mathsf{Set}(G)$ as functor categories.

A.4.2 The Yoneda embedding

The example $\mathsf{Graph} = \mathsf{Set}^{\cdot \rightrightarrows \cdot}$ leads one to wonder which categories $\mathcal{C}$ can be represented as functor categories $\mathsf{Set}^{\mathcal{D}}$ for a suitably chosen $\mathcal{D}$ or, when that is not possible, at least as full subcategories of $\mathsf{Set}^{\mathcal{D}}$.

For a locally small category $\mathcal{C}$, there is the hom-functor

$$\mathcal{C}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathsf{Set} .$$

By transposing as in (A.6) we obtain the functor

$$\mathsf{y} : \mathcal{C} \to \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$$

which maps an object $A \in \mathcal{C}$ to the representable functor

$$\mathsf{y}A = \mathcal{C}(-, A) : B \mapsto \mathcal{C}(B, A)$$

and a morphism $f : A \to A'$ in $\mathcal{C}$ to the natural transformation $\mathsf{y}f : \mathsf{y}A \Longrightarrow \mathsf{y}A'$ whose component at B is

$$(\mathsf{y}f)_B = \mathcal{C}(B, f) : g \mapsto f \circ g .$$

This functor y is called the *Yoneda embedding*.

Exercise A.4.4. Show that this *is* a functor.

Theorem A.4.5 (Yoneda embedding)**.** *For any locally small category $\mathcal{C}$ the Yoneda embedding*

$$\mathsf{y} : \mathcal{C} \to \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$$

is full and faithful and injective on objects. Therefore, $\mathcal{C}$ is a full subcategory of $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.

The proof of the theorem uses the famous Yoneda Lemma.

Lemma A.4.6 (Yoneda)**.** *Every functor $F : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$ is naturally isomorphic to the functor $\mathsf{Nat}(\mathsf{y}-, F)$. That is, for every $A \in \mathcal{C}$,*

$$\mathsf{Nat}(\mathsf{y}A, F) \cong FA \ ,$$

and this isomorphism is natural in A.

Indeed, the displayed isomorphism is also natural in F.

Proof. The desired natural isomorphism θ_A maps a natural transformation $\eta \in \mathsf{Nat}(\mathsf{y}A, F)$ to $\eta_A 1_A$. The inverse ${\theta_A}^{-1}$ maps an element $x \in FA$ to the natural transformation $({\theta_A}^{-1}x)$ whose component at B maps $f \in \mathcal{C}(B, A)$ to $(Ff)x$. To summarize, for $\eta : \mathcal{C}(-, A) \Longrightarrow F$, $x \in FA$ and $f \in \mathcal{C}(B, A)$, we have

$$\begin{aligned} &\theta_A : \mathsf{Nat}(\mathsf{y}A, F) \to FA \ , && {\theta_A}^{-1} : FA \to \mathsf{Nat}(\mathsf{y}A, F) \ , \\ &\theta_A \eta = \eta_A 1_A \ , && ({\theta_A}^{-1}x)_B f = (Ff)x \ . \end{aligned}$$

To see that θ_A and ${\theta_A}^{-1}$ really are inverses of each other, observe that

$$\theta_A({\theta_A}^{-1}x) = ({\theta_A}^{-1}x)_A 1_A = (F1_A)x = 1_{FA}x = x \ ,$$

and also

$$({\theta_A}^{-1}(\theta_A \eta))_B f = (Ff)(\theta_A \eta) = (Ff)(\eta_A 1_A) = \eta_B(1_A \circ f) = \eta_B f \ ,$$

where the third equality holds by the following naturality square for η:

$$\begin{tikzcd}
\mathcal{C}(A, A) \arrow[r, "\eta_A"] \arrow[d, "{\mathcal{C}(f, A)}"'] & FA \arrow[d, "Ff"] \\
\mathcal{C}(B, A) \arrow[r, "\eta_B"'] & FB
\end{tikzcd}$$

It remains to check that θ is natural, which amounts to establishing the commutativity of the following diagram, with $g : A \to A'$:

$$\begin{tikzcd}
\mathsf{Nat}(\mathsf{y}A, F) \arrow[r, "\theta_A"] & FA \\
\mathsf{Nat}(\mathsf{y}A', F) \arrow[r, "\theta_{A'}"'] \arrow[u, "{\mathsf{Nat}(\mathsf{y}g, F)}"] & FA' \arrow[u, "Fg"']
\end{tikzcd}$$

The diagram is commutative because, for any $\eta : \mathsf{y}A' \Longrightarrow F$,

$$(Fg)(\theta_{A'}\eta) = (Fg)(\eta_{A'}1_{A'}) = \eta_A(1_{A'} \circ g) = \\ \eta_A(g \circ 1_A) = (\mathsf{Nat}(\mathsf{y}g, F)\eta)_A 1_A = \theta_A(\mathsf{Nat}(\mathsf{y}g, F)\eta) \ ,$$

where the second equality is justified by naturality of η. □

Proof of Theorem A.4.5. That the Yoneda embedding is full and faithful means that for all $A, B \in \mathcal{C}$ the map

$$\mathsf{y} : \mathcal{C}(A, B) \to \mathsf{Nat}(\mathsf{y}A, \mathsf{y}B)$$

which maps $f : A \to B$ to $\mathsf{y}f : \mathsf{y}A \Longrightarrow \mathsf{y}B$ is an isomorphism. But this is just the Yoneda Lemma applied to the case $F = \mathsf{y}B$. Indeed, with notation as in the proof of the Yoneda Lemma and $g : C \to A$, we see that the isomorphism

$$\theta_A^{-1} : \mathcal{C}(A, B) = (\mathsf{y}B)A \to \mathsf{Nat}(\mathsf{y}A, \mathsf{y}B)$$

is in fact y:

$$({\theta_A}^{-1}f)_C g = ((\mathsf{y}A)g)f = f \circ g = (\mathsf{y}f)_C g \ .$$

Furthermore, if $\mathsf{y}A = \mathsf{y}B$ then $1_A \in \mathcal{C}(A, A) = (\mathsf{y}A)A = (\mathsf{y}B)A = \mathcal{C}(B, A)$ which can only happen if $A = B$. Therefore, y is injective on objects. □

The following corollary is often useful.

Corollary A.4.7. *For $A, B \in \mathcal{C}$, $A \cong B$ if, and only if, $\mathsf{y}A \cong \mathsf{y}B$ in $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.*

Proof. Every functor preserves isomorphisms, and a full and faithful one also reflects them. (A functor $F : \mathcal{C} \to \mathcal{D}$ is said to *reflect* isomorphisms when $Ff : FA \to FB$ being an isomorphisms implies that $f : A \to B$ is an isomorphism.) □

Exercise A.4.8. Prove that a full and faithful functor reflects isomorphisms.

Functor categories $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$ are important enough to deserve a name. They are called *presheaf categories*, and a functor $F : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$ is called a *presheaf* on $\mathcal{C}$. We also use the notation $\widehat{\mathcal{C}} = \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.

A.4.3 Equivalence of categories

An isomorphism of categories $\mathcal{C}$ and $\mathcal{D}$ in Cat consists of functors

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "F"] & \mathcal{D} \arrow[l, bend left, "G"]
\end{tikzcd}$$

such that $G \circ F = 1_{\mathcal{C}}$ and $F \circ G = 1_{\mathcal{D}}$. This is often too restrictive a notion. A more general notion which replaces the above identities with natural isomorphisms is more useful.

Definition A.4.9. An *equivalence of categories* is a pair of functors

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

such that there are natural isomorphisms

$$G \circ F \cong 1_{\mathcal{C}} \qquad \text{and} \qquad F \circ G \cong 1_{\mathcal{D}} \; .$$

We say that $\mathcal{C}$ and $\mathcal{D}$ are *equivalent categories* and write $\mathcal{C} \simeq \mathcal{D}$.

A functor $F : \mathcal{C} \to \mathcal{D}$ is called an *equivalence functor* if there exists $G : \mathcal{D} \to \mathcal{C}$ such that F and G form an equivalence.

The point of equivalence of categories is that it preserves almost all categorical properties, but ignores those concepts that are not of interest from a categorical point of view, such as identity of objects.

The following proposition requires the Axiom of Choice as stated. However, in many specific cases a canonical choice can be made without appeal to that axiom.

Proposition A.4.10. *A functor $F : \mathcal{C} \to \mathcal{D}$ is an equivalence functor if, and only if, F is full and faithful, and* essentially surjective *on objects, meaning that for every $B \in \mathcal{D}$ there exists $A \in \mathcal{C}$ such that $FA \cong B$.*

Proof. It is easily seen that the conditions are necessary, so we only show they are sufficient. Suppose $F : \mathcal{C} \to \mathcal{D}$ is full and faithful, and essentially surjective on objects. For each $B \in \mathcal{D}$, *choose* an object $GB \in \mathcal{C}$ and an isomorphism $\eta_B : F(GB) \to B$. If $f : B \to C$ is a morphism in $\mathcal{D}$, let $Gf : GB \to GC$ be the unique morphism in $\mathcal{C}$ for which

$$F(Gf) = {\eta_C}^{-1} \circ f \circ \eta_B \; . \tag{A.7}$$

Such a unique morphism exists because F is full and faithful. This defines a functor $G : \mathcal{D} \to \mathcal{C}$, as can be easily checked. In addition, (A.7) ensures that η is a natural isomorphism $F \circ G \Longrightarrow 1_{\mathcal{D}}$.

It remains to show that $G \circ F \cong 1_{\mathcal{C}}$. For $A \in \mathcal{C}$, let $\theta_A : G(FA) \to A$ be the unique morphism such that $F\theta_A = \eta_{FA}$. Naturality of θ_A follows from functoriality of F and naturality of η. Because F reflects isomorphisms, θ_A is an isomorphism for every A. □

Example A.4.11. As an example of equivalence of categories we consider the category of sets and partial functions and the category of pointed sets.

A *partial function* $f : A \rightharpoonup B$ is a function defined on a subset $\mathsf{supp}\, f \subseteq A$, called the *support*[3] of f, and taking values in B. Composition of partial functions $f : A \rightharpoonup B$ and $g : B \rightharpoonup C$ is the partial function $g \circ f : A \rightharpoonup C$ defined by

$$\begin{aligned} \mathsf{supp}\,(g \circ f) &= \big\{ x \in A \;\big|\; x \in \mathsf{supp}\, f \wedge fx \in \mathsf{supp}\, g \big\} \\ (g \circ f)x &= g(fx) \quad \text{for } x \in \mathsf{supp}\,(g \circ f) \end{aligned}$$

[3] The support of a partial function $f : A \rightharpoonup B$ is usually called its *domain*, but this terminology conflicts with A being the domain of f as a morphism.

Composition of partial functions is associative. This way we obtain a category Par of sets and partial functions.

A *pointed set* (A, a) is a set A together with an element $a \in A$. A *pointed function* $f : (A, a) \to (B, b)$ between pointed sets is a function $f : A \to B$ such that $fa = b$. The category $\mathsf{Set}_\bullet$ consists of pointed sets and pointed functions.

The categories Par and $\mathsf{Set}_\bullet$ are equivalent. The equivalence functor $F : \mathsf{Set}_\bullet \to \mathsf{Par}$ maps a pointed set (A, a) to the set $F(A, a) = A \setminus \{a\}$, and a pointed function $f : (A, a) \to (B, b)$ to the partial function $Ff : F(A, a) \rightharpoonup F(B, b)$ defined by

$$\mathsf{supp}\,(Ff) = \left\{x \in A \mid fx \neq b\right\} , \qquad\qquad (Ff)x = fx .$$

The inverse equivalence functor $G : \mathsf{Par} \to \mathsf{Set}_\bullet$ maps a set $A \in \mathsf{Par}$ to the pointed set $GA = (A + \{\bot_A\}, \bot_A)$, where $\bot_A$ is an element that does not belong to A. A partial function $f : A \rightharpoonup B$ is mapped to the pointed function $Gf : GA \to GB$ defined by

$$(Gf)x = \begin{cases} fx & \text{if } x \in \mathsf{supp}\, f \\ \bot_B & \text{otherwise} . \end{cases}$$

A good way to think about the “bottom” point $\bot_A$ is as a special “undefined value”. Let us look at the composition of F and G on objects:

$$\begin{aligned} G(F(A, a)) &= G(A \setminus \{a\}) = ((A \setminus \{a\}) + \bot_A, \bot_A) \cong (A, a) . \\ F(GA) &= F(A + \{\bot_A\}, \bot_A) = (A + \{\bot_A\}) \setminus \{\bot_A\} = A . \end{aligned}$$

The isomorphism $G(F(A, a)) \cong (A, a)$ is easily seen to be natural.

Example A.4.12. Another example of an equivalence of categories arises when we take the poset reflection of a preorder. Let $(P, \leq)$ be a preorder, If we think of P as a category, then $a, b \in P$ are isomorphic, when $a \leq b$ and $b \leq a$. Isomorphism $\cong$ is an equivalence relation, therefore we may form the quotient set $P/{\cong}$. The set $P/{\cong}$ is a poset for the order relation $\sqsubseteq$ defined by

$$[a] \sqsubseteq [b] \iff a \leq b .$$

Here $[a]$ denotes the equivalence class of a. We call $(P/{\cong}, \sqsubseteq)$ the *poset reflection* of P. The quotient map $q : P \to P/{\cong}$ is a functor when P and $P/{\cong}$ are viewed as categories. By Proposition A.4.10, q is an equivalence functor. Trivially, it is faithful and surjective on objects. It is also full because $qa \sqsubseteq qb$ in $P/{\cong}$ implies $a \leq b$ in P.

A.5 Adjoint Functors

The notion of adjunction is perhaps the most important concept revealed by category theory. It is a fundamental logical and mathematical concept that occurs everywhere and often marks an important and interesting connection between two constructions of interest. In logic, adjoint functors are pervasive, although this is only recognizable through the lens of category theory.

A.5.1 Adjoint maps between preorders

Let us begin with a simple situation. We have already seen that a preorder $(P, \leq)$ is a category in which there is at most one morphism between any two objects. A functor between preorders is a monotone map. Suppose we have preorders P and Q with monotone maps back and forth,

$$\begin{tikzcd} P \arrow[r, bend left, "f"] & Q \arrow[l, bend left, "g"] \end{tikzcd} .$$

We say that f and g are *adjoint*, and write $f \dashv g$, when for all $x \in P$, $y \in Q$,

$$fx \leq y \iff x \leq gy . \tag{A.8}$$

Note that adjointness is *not* a symmetric relation. The map f is the *left adjoint* and g is the *right adjoint* (note their positions with respect to $\leq$).

Equivalence (A.8) is more conveniently displayed as

$$\begin{array}{c} fx \leq y \\ \hline\hline x \leq gy \end{array}$$

The double line indicates the fact that this is a two-way rule: the top line implies the bottom line, and vice versa.

Let us consider two examples.

Conjunction is adjoint to implication Consider a propositional calculus with logical operations of conjunction $\wedge$ and implication $\Rightarrow$ (perhaps among others). The formulas of this calculus are built from variables x_0, x_1, x_2, …, the truth values $\bot$ and $\top$, and the logical connectives $\wedge, \Rightarrow, \ldots$. The logical rules are given in natural deduction style:

$$\frac{}{\top} \qquad\qquad \frac{\bot}{A} \qquad\qquad \frac{A \qquad B}{A \wedge B} \qquad\qquad \frac{A \wedge B}{A} \qquad\qquad \frac{A \wedge B}{B}$$

$$\frac{A \Rightarrow B \qquad A}{B} \qquad\qquad \begin{array}{c} [u : A] \\ \vdots \\ B \\ \hline A \Rightarrow B \end{array} u$$

For example, we read the inference rules for $\Rightarrow$ as, respectively, "from $A \Rightarrow B$ and A we infer B" and "if from assumption A we infer B, then (without any assumptions) we infer $A \Rightarrow B$". Discharged assumptions are indicated by enclosing them in brackets, along with a label $[u : A]$ for the assumption, which is recorded along with the rule that discharges it, as above.

Logical entailment $\vdash$ between formulas of the propositional calculus is the relation $A \vdash B$ which holds if, and only if, from assuming A we can infer B (by using only the inference rules of the calculus). It is trivially the case that $A \vdash A$, and also

$$\text{if } A \vdash B \text{ and } B \vdash C \text{ then } A \vdash C \ .$$

In other words, $\vdash$ is a reflexive and transitive relation on the set P of all propositional formulas, so that $(\mathsf{P}, \vdash)$ is a preorder.

Let A be a propositional formula. Define $f : \mathsf{P} \to \mathsf{P}$ and $g : \mathsf{P} \to \mathsf{P}$ to be the maps

$$fB = (A \wedge B) \ , \qquad\qquad gB = (A \Rightarrow B) \ .$$

To see that the maps f and g are functors we need to show they respect entailment. Indeed, if $B \vdash B'$ then $A \wedge B \vdash A \wedge B'$ and $A \Rightarrow B \vdash A \Rightarrow B'$ by the following two derivations.

$$\dfrac{\dfrac{A \wedge B}{A} \qquad \begin{matrix}\dfrac{A \wedge B}{B} \\ \vdots \\ B'\end{matrix}}{A \wedge B'} \qquad\qquad \dfrac{\begin{matrix}\dfrac{A \Rightarrow B \qquad [u : A]}{B} \\ \vdots \\ B'\end{matrix}}{A \Rightarrow B'}\,u$$

We claim that $f \dashv g$. For this we need to prove that $A \wedge B \vdash C$ if, and only if, $B \vdash A \Rightarrow C$. The following two derivations establish the required equivalence.

$$\dfrac{\begin{matrix}\dfrac{[u : A] \qquad B}{A \wedge B} \\ \vdots \\ C\end{matrix}}{A \Rightarrow C}\,u \qquad\qquad \dfrac{\begin{matrix}\dfrac{A \wedge B}{B} \\ \vdots \\ A \Rightarrow C\end{matrix} \qquad \dfrac{A \wedge B}{A}}{C}$$

Therefore, *conjunction is left adjoint to implication.*

Topological interior as an adjoint Recall that a *topological space* $(X, \mathcal{O}X)$ is a set X together with a family $\mathcal{O}X \subseteq \mathcal{P}X$ of subsets of X which contains $\emptyset$ and X, and is closed under finite intersections and arbitrary unions. The elements of $\mathcal{O}X$ are called the *open sets.*

The *topological interior* of a subset $S \subseteq X$ is the largest open set contained in S, namely,

$$\mathsf{int}\, S = \bigcup \left\{ U \in \mathcal{O}X \mid U \subseteq S \right\} \ .$$

Both $\mathcal{O}X$ and $\mathcal{P}X$ are posets ordered by subset inclusion. The inclusion $i : \mathcal{O}X \to \mathcal{P}X$ is thus a monotone map, and so indeed is the interior $\mathsf{int} : \mathcal{P}X \to \mathcal{O}X$, as follows immediately from its construction. So we have:

$$\begin{tikzcd} \mathcal{O}X \arrow[r, bend left, "i"] & \mathcal{P}X \arrow[l, bend left, "\mathsf{int}"] \end{tikzcd}$$

Moreover, for $U \in \mathcal{O}X$ and $S \in \mathcal{P}X$ we plainly also have

$$\frac{iU \subseteq S}{\overline{U \subseteq \mathsf{int}\, S}}$$

since $\mathsf{int}\, S$ is the largest open set contained in S. Thus *topological interior is right adjoint* to the inclusion of $\mathcal{O}X$ into $\mathcal{P}X$.

A.5.2 Adjoint functors

Let us now generalize the notion of adjoint monotone maps from posets to the situation

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

with arbitrary categories and functors. For monotone maps $f \dashv g$, the adjunction condition is a bijection

$$\frac{fx \to y}{\overline{x \to gy}}$$

between morphisms of the form $fx \to y$ and morphisms of the form $x \to gy$. This is the notion that generalizes the special case; for any $A \in \mathcal{C}$, $B \in \mathcal{D}$ we require a bijection between the sets $\mathcal{D}(FA, B)$ and $\mathcal{C}(A, GB)$:

$$\frac{FA \to B}{\overline{A \to GB}}$$

Definition A.5.1. An *adjunction* $F \dashv G$ between the functors

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

is a natural isomorphism θ between functors

$$\mathcal{D}(F-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathsf{Set} \qquad \text{and} \qquad \mathcal{C}(-,G-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathsf{Set}\ .$$

This means that for every $A \in \mathcal{C}$ and $B \in \mathcal{D}$ there is a bijection

$$\theta_{A,B} : \mathcal{D}(FA, B) \cong \mathcal{C}(A, GB)\ ,$$

and naturality of θ means that for $f : A' \to A$ in $\mathcal{C}$ and $g : B \to B'$ in $\mathcal{D}$ the following diagram commutes:

$$\begin{tikzcd} \mathcal{D}(FA, B) \arrow[r, "\theta_{A,B}"] \arrow[d, "\mathcal{D}(Ff, g)"'] & \mathcal{D}(A, GB) \arrow[d, "\mathcal{C}(f, Gg)"] \\ \mathcal{D}(FA', B') \arrow[r, "\theta_{A',B'}"'] & \mathcal{C}(A', GB') \end{tikzcd}$$

Equivalently, for every $h : FA \to B$ in $\mathcal{D}$,

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

We say that F is the *left adjoint* and G is the *right adjoint*.

We have already seen examples of adjoint functors. For any category $\mathbb{B}$ we have functors $(-) \times \mathbb{B}$ and $(-)^{\mathbb{B}}$ from Cat to Cat. Recall the isomorphism (A.6),

$$\mathsf{Cat}(\mathbb{A} \times \mathbb{B}, \mathbb{C}) \cong \mathsf{Cat}(\mathbb{A}, \mathbb{C}^{\mathbb{B}}) \ .$$

This isomorphism is in fact natural in $\mathbb{A}$ and $\mathbb{C}$, so that

$$(-) \times \mathbb{B} \dashv (-)^{\mathbb{B}} \ .$$

Similarly, for any set $B \in \mathsf{Set}$ there are functors

$$(-) \times B : \mathsf{Set} \to \mathsf{Set} \ , \qquad\qquad (-)^{B} : \mathsf{Set} \to \mathsf{Set} \ ,$$

where $A \times B$ is the cartesian product of A and B, and C^B is the set of all functions from B to C. For morphisms, $f \times B = f \times 1_B$ and $f^B = f \circ (-)$. We then indeed have a natural isomorphism, for all $A, C \in \mathsf{Set}$,

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

which maps a function $f : A \times B \to C$ to the function $(\widetilde{f}x)y = f\langle x, y \rangle$. Therefore,

$$(-) \times B \dashv (-)^{B} \ .$$

Exercise A.5.2. Verify that the definition (A.8) of adjoint monotone maps between preorders is a special case of Definition A.5.1. What happened to the naturality condition?

For another example, consider the forgetful functor

$$U : \mathsf{Cat} \to \mathsf{Graph} \ ,$$

which maps a category to the underlying directed graph. It has a left adjoint $P \dashv U$. The functor P is the *free* construction of a category from a graph; it maps a graph G to the *category of paths* $P(G)$. The objects of $P(G)$ are the vertices of G. The morphisms of $P(G)$ are the finite paths

$$\begin{tikzcd} v_0 \arrow[r, "e_1"] & v_1 \arrow[r, "e_2"] & \cdots \arrow[r, "e_n"] & v_n \end{tikzcd}$$

of edges in G, composition is concatenation of paths, and the identity morphism on a vertex v is the empty path starting and ending at v.

By using the Yoneda Lemma we can easily prove that adjoints are unique up to natural isomorphism.

Proposition A.5.3. *Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be adjoint functors, with $F \dashv G$. If also $G' : \mathcal{D} \to \mathcal{C}$ with $F \dashv G'$, then $G \cong G'$.*

Proof. Since the Yoneda embedding is full and faithful, we have $GB \cong G'B$ if, and only if, $\mathcal{C}(-, GB) \cong \mathcal{C}(-, G'B)$. But this indeed holds, because, for any $A \in \mathcal{C}$, we have

$$\mathcal{C}(A, GB) \cong \mathcal{D}(FA, B) \cong \mathcal{C}(A, G'B) ,$$

naturally in A. □

Left adjoints are of course also unique up to isomorphism, by duality.

A.5.3 The unit of an adjunction

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be adjoint functors, $F \dashv G$, and let $\theta : \mathcal{D}(F-, -) \to \mathcal{C}(-, G-)$ be the natural isomorphism witnessing the adjunction. For any object $A \in \mathcal{C}$ there is a distinguished morphism $\eta_A = \theta_{A,FA} 1_{FA} : A \to G(FA)$,

$$\frac{1_{FA} : FA \to FA}{\eta_A : A \to G(FA)}$$

Since θ is natural in A, we have a natural transformation $\eta : 1_{\mathcal{C}} \Longrightarrow G \circ F$, which is called the *unit of the adjunction* $F \dashv G$. In fact, we can recover θ from η as follows. For $f : FA \to B$, we have

$$\theta_{A,B} f = \theta_{A,B}(f \circ 1_{FA}) = Gf \circ \theta_{A,FA}(1_{FA}) = Gf \circ \eta_A ,$$

where we used naturality of θ in the second step. Schematically, given any $f : FA \to B$, the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[dr, "\theta_{A,B} f"'] & G(FA) \arrow[d, "Gf"] \\
& GB
\end{tikzcd}$$

Since $\theta_{A,B}$ is a bijection, it follows that *every* morphism $g : A \to GB$ has the form $g = Gf \circ \eta_A$ for a *unique* $f : FA \to B$. We say that $\eta_A : A \to G(FA)$ is a *universal* morphism to G, or that η has the following *universal mapping property*: for every $A \in \mathcal{C}$, $B \in \mathcal{D}$, and $g : A \to GB$, there exists a *unique* $f : FA \to B$ such that $g = Gf \circ \eta_A$:

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[dr, "g"'] & G(FA) \arrow[d, "Gf"] & FA \arrow[d, dotted, "f"] \\
& GB & B
\end{tikzcd}$$

This means that an adjunction can be given in terms of its unit. The isomorphism $\theta : \mathcal{D}(F-,-) \to \mathcal{C}(-,G-)$ is then recovered by

$$\theta_{A,B} f = Gf \circ \eta_A \; .$$

Proposition A.5.4. *A functor $F : \mathcal{C} \to \mathcal{D}$ is left adjoint to a functor $G : \mathcal{D} \to \mathcal{C}$ if, and only if, there exists a natural transformation*

$$\eta : 1_{\mathcal{C}} \Longrightarrow G \circ F \; ,$$

called the unit *of the adjunction, such that, for all $A \in \mathcal{C}$ and $B \in \mathcal{D}$ the map $\theta_{A,B} : \mathcal{D}(FA, B) \to \mathcal{C}(A, GB)$, defined by*

$$\theta_{A,B} f = Gf \circ \eta_A \; ,$$

is an isomorphism.

Let us demonstrate how the universal mapping property of the unit of an adjunction appears as a well known construction in algebra. Consider the forgetful functor from monoids to sets,

$$U : \mathsf{Mon} \to \mathsf{Set} \; .$$

Does it have a left adjoint $F : \mathsf{Set} \to \mathsf{Mon}$? In order to obtain one, we need a "most economical" way of making a monoid FX from a given set X. Such a construction readily suggests itself, namely the *free monoid* on X, consisting of finite sequences of elements of X,

$$FX = \left\{ x_1 \dots x_n \;\middle|\; n \geq 0 \;\&\; x_1, \dots, x_n \in X \right\} \; .$$

The monoid operation is concatenation of sequences

$$x_1 \dots x_m \cdot y_1 \dots y_n = x_1 \dots x_m y_1 \dots y_n \; ,$$

and the empty sequence is the unit of the monoid. In order for F to be a functor, it should also map morphisms to morphisms. If $f : X \to Y$ is a function, define $Ff : FX \to FY$ by

$$Ff : x_1 \dots x_n \mapsto (fx_1) \dots (fx_n) \; .$$

There is an inclusion $\eta_X : X \to U(FX)$ which maps every element $x \in X$ to the singleton sequence x. This gives a natural transformation $\eta : 1_{\mathsf{Set}} \Longrightarrow U \circ F$.

The monoid FX is "free" in the sense that it "satisfies only the equations required by the monoid laws"; we make this precise as follows. For every monoid M and function $f : X \to UM$ there exists a unique monoid homomorphism $\overline{f} : FX \to M$ such that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "f"'] & U(FX) \arrow[d, "U\overline{f}"] \\
& UM
\end{tikzcd}$$

This is precisely the condition required by Proposition A.5.4 for η to be the unit of the adjunction $F \dashv U$. In this case, the universal mapping property of η is just the usual characterization of the free monoid FX generated by the set X: a homomorphism from FX is uniquely determined by its values on the generators.

A.5.4 The counit of an adjunction

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be adjoint functors with $F \dashv G$, and let $\theta : \mathcal{D}(F-,-) \to \mathcal{C}(-,G-)$ be the natural isomorphism witnessing the adjunction. For any object $B \in \mathcal{D}$ we have a distinguished morphism $\varepsilon_B = \theta^{-1}_{GB,B} 1_{GB} : F(GB) \to B$ by:

$$\begin{array}{c} 1_{GB} : GB \to GB \\ \hline\hline \varepsilon_B : F(GB) \to B \end{array}$$

The natural transformation $\varepsilon : F \circ G \Longrightarrow 1_{\mathcal{D}}$ is called the *counit* of the adjunction $F \dashv G$. It is the dual notion to the unit of an adjunction. We state briefly the basic properties of the counit, which are easily obtained by "turning around" all the morphisms in the previous section and exchanging the roles of the left and right adjoints.

The bijection $\theta^{-1}_{A,B}$ can be recovered from the counit. For $g : A \to GB$ in $\mathcal{C}$, we have

$$\theta^{-1}_{A,B} g = \theta^{-1}_{A,B}(1_{GB} \circ g) = \theta^{-1}_{A,B} 1_{GB} \circ Fg = \varepsilon_B \circ Fg \ .$$

The universal mapping property of the counit is this: for every $A \in \mathcal{C}$, $B \in \mathcal{D}$, and $f : FA \to B$, there exists a *unique* $g : A \to GB$ such that $f = \varepsilon_B \circ Fg$:

$$\begin{tikzcd}
B & F(GB) \arrow[l, "\varepsilon_B"'] & & GB \\
& FA \arrow[ul, "f"] \arrow[u, "Fg"'] & & A \arrow[u, dotted, "g"']
\end{tikzcd}$$

The following is the dual of Proposition A.5.4.

Proposition A.5.5. *A functor $F : \mathcal{C} \to \mathcal{D}$ is left adjoint to a functor $G : \mathcal{D} \to \mathcal{C}$ if, and only if, there exists a natural transformation*

$$\varepsilon : F \circ G \Longrightarrow 1_{\mathcal{D}} \ ,$$

called the counit *of the adjunction, such that, for all $A \in \mathcal{C}$ and $B \in \mathcal{D}$ the map $\theta^{-1}_{A,B} : \mathcal{C}(A, GB) \to \mathcal{D}(FA, B)$, defined by*

$$\theta^{-1}_{A,B} g = \varepsilon_B \circ Fg \ ,$$

is an isomorphism.

Let us consider again the forgetful functor $U : \mathsf{Mon} \to \mathsf{Set}$ and its left adjoint $F : \mathsf{Set} \to \mathsf{Mon}$, the free monoid construction. For a monoid $(M, \star) \in \mathsf{Mon}$, the counit of the adjunction $F \dashv U$ is a monoid homomorphism $\varepsilon_M : F(UM) \to M$, defined by

$$\varepsilon_M(x_1 x_2 \ldots x_n) = x_1 \star x_2 \star \cdots \star x_n \ .$$

It has the following universal mapping property: for $X \in \mathsf{Set}$, $(M, \star) \in \mathsf{Mon}$, and a homomorphism $f : FX \to M$ there exists a unique function $\overline{f} : X \to UM$ such that $f = \varepsilon_M \circ F\overline{f}$, namely

$$\overline{f}x = fx \ ,$$

where in the above definition $x \in X$ is viewed as an element of the set X on the left-hand side, and as an element of the free monoid FX on the right-hand side. To summarize, the universal mapping property of the counit ε is the familiar piece of wisdom that a homomorphism $f : FX \to M$ from a free monoid is already determined by its values on the generators.

A.6 Limits and Colimits

The following limits and colimits are all special cases of adjoint functors, as we shall see.

A.6.1 Binary products

In a category $\mathcal{C}$, the *(binary) product* of objects A and B is an object $A \times B$ together with *projections* $\pi_0 : A \times B \to A$ and $\pi_1 : A \times B \to B$ such that, for every object $C \in \mathcal{C}$ and every pair of morphisms $f : C \to A$, $g : C \to B$ there exists a *unique* morphism $h : C \to A \times B$ for which the following diagram commutes:

$$\begin{tikzcd}
& C \arrow[ld, "f"'] \arrow[d, dotted, "h"] \arrow[rd, "g"] & \\
A & A \times B \arrow[l, "\pi_0"] \arrow[r, "\pi_1"'] & B
\end{tikzcd}$$

We normally refer to the product $(A \times B, \pi_0, \pi_1)$ just by its object $A \times B$, but you should keep in mind that a product is given by an object *and* two projections. The arrow $h : C \to A \times B$ is denoted by $\langle f, g \rangle$. The property

$$\begin{aligned}
&\text{for all } C, \text{ for all } f : C \to A, \text{ for all } g : C \to B, \\
&\text{there is a unique } h : C \to A \times B, \\
&\text{with } \pi_0 \circ h = f \ \& \ \pi_1 \circ h = g
\end{aligned}$$

is the *universal mapping property* of the product $A \times B$. It characterizes the product of A and B uniquely up to isomorphism in the sense that if $(P, p_0 : P \to A, p_1 : P \to B)$ is another product of A and B, then there is a unique isomorphism $r : P \xrightarrow{\sim} A \times B$ such that $p_0 = \pi_0 \circ r$ and $p_1 = \pi_1 \circ r$.

If in a category $\mathcal{C}$ every two objects have a product, we can turn binary products into an operation[4] by *choosing* a product $A \times B$ for each pair of objects $A, B \in \mathcal{C}$. In general this requires the Axiom of Choice, but in many specific cases a particular choice of products can be made without appeal to that axiom. When we view binary products as an operation, we say that "$\mathcal{C}$ has chosen products". The same holds for other instances of limits and colimits.

For example, in Set the usual cartesian product of sets is a product. In categories of structures, products are the usual construction: the product of topological spaces in Top is their topological product, the product of directed graphs in Graph is their cartesian product, the product of categories in Cat is their product category, and so on.

A.6.2 Terminal objects

A *terminal object* in a category $\mathcal{C}$ is an object $\mathsf{1} \in \mathcal{C}$ such that for every $A \in \mathcal{C}$ there exists a *unique* morphism $!_A : A \to \mathsf{1}$.

For example, in Set an object is terminal if, and only if, it is a singleton. The terminal object in Cat is the unit category $\mathsf{1}$ consisting of one object and one morphism.

Exercise A.6.1. Prove that if $\mathsf{1}$ and $\mathsf{1}'$ are terminal objects in a category then they are isomorphic.

Exercise A.6.2. Let Field be the category whose objects are fields and morphisms are field homomorphisms.[5] Does Field have a terminal object? What about the category Ring of rings?

A.6.3 Equalizers

Given objects and morphisms

$$\begin{tikzcd} E \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

we say that e *equalizes* f and g when $f \circ e = g \circ e$.[6] An *equalizer* of f and g is a *universal* equalizing morphism; thus $e : E \to A$ is an equalizer of f and g when it equalizes them and, for all $k : K \to A$, if $f \circ k = g \circ k$ then there exists a unique morphism $m : K \to E$

[4] More precisely, binary product is a functor from $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$, cf. Section A.6.11.

[5] A field $(F, +, \cdot, ^{-1}, 0, 1)$ is a ring with a unit in which all non-zero elements have inverses. We also require that $0 \neq 1$. A homomorphism of fields preserves addition and multiplication, and consequently also 0, 1 and inverses.

[6] Note that this does *not* mean the diagram involving f, g and e is commutative!

such that $k = e \circ m$:

$$\begin{tikzcd}
E \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \\
K \arrow[u, dotted, "m"] \arrow[ru, "k"'] & &
\end{tikzcd}$$

In Set the equalizer of parallel functions $f : A \to B$ and $g : A \to B$ is the set

$$E = \{x \in A \mid fx = gx\}$$

with $e : E \to A$ being the subset inclusion $E \subseteq A$, $ex = x$. In general, equalizers can be thought of as those subobjects (subsets, subgroups, subspaces, ...) that can be defined by an equation.

Exercise A.6.3. Show that an equalizer is a monomorphism, i.e., if $e : E \to A$ is an equalizer of f and g, then, for all $r, s : C \to E$, $e \circ r = e \circ s$ implies $r = s$.

Definition A.6.4. A morphism is a *regular mono* if it is an equalizer.

The difference between monos and regular monos is best illustrated in the category Top: a continuous map $f : X \to Y$ is mono when it is injective, whereas it is a regular mono when it is a topological embedding.[7]

A.6.4 Pullbacks

A *pullback* of $f : A \to C$ and $g : B \to C$ is an object P with morphisms $p_0 : P \to A$ and $p_1 : P \to B$ such that $f \circ p_0 = g \circ p_1$, and whenever Q, $q_0 : Q \to A$, and $q_1 : Q \to B$ are such that $f \circ q_0 = g \circ q_1$, there then exists a unique $h : Q \to P$ such that $q_0 = p_0 \circ h$ and $q_1 = p_1 \circ h$:

$$\begin{tikzcd}
Q \arrow[rrd, bend left, "q_1"] \arrow[rdd, bend right, "q_0"'] \arrow[rd, dotted, "h"] & & \\
& P \arrow[r, "p_1"] \arrow[d, "p_0"'] \arrow[rd, phantom, "\lrcorner", very near start] & B \arrow[d, "g"] \\
& A \arrow[r, "f"'] & C
\end{tikzcd}$$

We indicate that P is a pullback by drawing a square corner next to it, as in the above diagram. The pullback is sometimes written $A \times_C B$, since it is indeed a product in the slice category over C.

[7] A continuous map $f : X \to Y$ is a topological embedding when, for every $U \in \mathcal{O}X$, the image $f[U]$ is an open subset of the image $\mathsf{im}(f)$; this means that there exists $V \in \mathcal{O}Y$ such that $f[U] = V \cap \mathsf{im}(f)$.

In Set, the pullback of $f : A \to C$ and $g : B \to C$ is the set

$$P = \big\{ \langle x, y \rangle \in A \times B \;\big|\; fx = gy \big\}$$

and the functions $p_0 : P \to A$, $p_1 : P \to B$ are the projections, $p_0\langle x, y\rangle = x$, $p_1\langle x, y\rangle = y$.

When we form the pullback of $f : A \to C$ and $g : B \to C$ we may also say that we *pull g back along f* and draw the diagram

$$\begin{tikzcd}
f^*B \arrow[r] \arrow[d, "f^*g"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "g"] \\
A \arrow[r, "f"'] & C
\end{tikzcd}$$

We think of $f^*g : f^*B \to A$ as the inverse image of B along f. This terminology is explained by looking at the pullback of a subset inclusion $u : U \hookrightarrow C$ along a function $f : A \to C$ in the category Set:

$$\begin{tikzcd}
f^*U \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & U \arrow[d, hook, "u"] \\
A \arrow[r, "f"'] & C
\end{tikzcd}$$

In this case the pullback is $\big\{ \langle x, y\rangle \in A \times U \;\big|\; fx = y \big\} \cong \big\{ x \in A \;\big|\; fx \in U \big\} = f^*U$, the inverse image of U along f.

Exercise A.6.5. Prove that in a category $\mathcal{C}$, a morphism $f : A \to B$ is mono if, and only if, the following diagram is a pullback:

$$\begin{tikzcd}
A \arrow[r, "\mathsf{1}_A"] \arrow[d, "\mathsf{1}_A"'] & A \arrow[d, "f"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

A.6.5 Limits

Let us now define the general notion of a limit.

A *diagram of shape $\mathcal{I}$* in a category $\mathcal{C}$ is a functor $D : \mathcal{I} \to \mathcal{C}$, where the category $\mathcal{I}$ is called the *index category*. We use letters i, j, k, … for objects of an index category $\mathcal{I}$, call them *indices*, and write D_i, D_j, D_k, … instead of Di, Dj, Dk, …

For example, if $\mathcal{I}$ is the category with three objects and three morphisms

$$\begin{tikzcd}
& 1 \arrow[dl, "12"'] \arrow[d, "13"] \\
2 \arrow[r, "23"'] & 3
\end{tikzcd}$$

where $13 = 23 \circ 12$ then a diagram of shape $\mathcal{I}$ is a commutative diagram

$$\begin{tikzcd}
& D_1 \arrow[dl, "d_{12}"'] \arrow[d, "d_{13}"] \\
D_2 \arrow[r, "d_{23}"'] & D_3
\end{tikzcd} \tag{A.9}$$

For each object $A \in \mathcal{C}$, the *constant A-valued diagram* of shape $\mathcal{I}$ is given by the constant functor $\Delta_A : \mathcal{I} \to \mathcal{C}$, which maps every object to A and every morphism to $\mathsf{1}_A$.

Let $D : \mathcal{I} \to \mathcal{C}$ be a diagram of shape $\mathcal{I}$. A *cone* on D from an object $A \in \mathcal{C}$ is a natural transformation $\alpha : \Delta_A \Longrightarrow D$. This means that for every index $i \in \mathcal{I}$ there is a morphism $\alpha_i : A \to D_i$ such that whenever $u : i \to j$ in $\mathcal{I}$ then $\alpha_j = Du \circ \alpha_i$.

For a given diagram $D : \mathcal{I} \to \mathcal{C}$, we can collect all cones on D into a category $\mathsf{Cone}(D)$ whose objects are cones on D. A morphism between cones $f : (A, \alpha) \to (B, \beta)$ is a morphism $f : A \to B$ in $\mathcal{C}$ such that $\alpha_i = \beta_i \circ f$ for all $i \in \mathcal{I}$. Morphisms in $\mathsf{Cone}(D)$ are composed as morphisms in $\mathcal{C}$. A morphism $f : (A, \alpha) \to (B, \beta)$ is also called a factorization of the cone (A, α) through the cone (B, β).

A *limit* of a diagram $D : \mathcal{I} \to \mathcal{C}$ is a terminal object in $\mathsf{Cone}(D)$. Explicitly, a limit of D is given by a cone (L, λ) such that for every other cone (A, α) there exists a *unique* morphism $f : A \to L$ such that $\alpha_i = \lambda_i \circ f$ for all $i \in \mathcal{I}$. We denote (the object part of) a limit of D by one of the following:

$$\lim D \qquad \lim_{i \in \mathcal{I}} D_i \qquad \varprojlim_{i \in \mathcal{I}} D_i\ .$$

Limits are also called *projective limits.* We say that a category *has limits of shape* $\mathcal{I}$ when every diagram of shape $\mathcal{I}$ in $\mathcal{C}$ has a limit.

Products, terminal objects, equalizers, and pullbacks are all special cases of limits:

- a product $A \times B$ is the limit of the functor $D : \mathsf{2} \to \mathcal{C}$ where $\mathsf{2}$ is the discrete category on two objects 0 and 1, and $D_0 = A$, $D_1 = B$.

- a terminal object $\mathsf{1}$ is the limit of the (unique) functor $D : \mathsf{0} \to \mathcal{C}$ from the empty category.

- an equalizer of $f, g : A \to B$ is the limit of the functor $D : (\cdot \rightrightarrows \cdot) \to \mathcal{C}$ which maps one morphism to f and the other one to g.

- the pullback of $f : A \to C$ and $g : B \to C$ is the limit of the functor $D : \mathcal{I} \to \mathcal{C}$ where $\mathcal{I}$ is the category

$$\begin{tikzcd}
 & \bullet \arrow[d, "2"] \\
\bullet \arrow[r, "1"'] & \bullet
\end{tikzcd}$$

  with $D1 = f$ and $D2 = g$.

It is clear how to define the product of an arbitrary family of objects

$$\{A_i \in \mathcal{C} \mid i \in I\} \ .$$

Such a family is a diagram of shape I, where I is viewed as a discrete category. A *product* $\prod_{i \in I} A_i$ is then given by an object $P \in \mathcal{C}$ and morphisms $\pi_i : P \to A_i$ such that, whenever we have a family of morphisms $\{f_i : B \to A_i \mid i \in I\}$ there exists a *unique* morphism $\langle f_i \rangle_{i \in I} : B \to P$ such that $f_i = \pi_i \circ f$ for all $i \in I$.

A *finite product* is a product of a finite family. As a special case we see that a terminal object is the product of an empty family. It is not hard to show that a category has finite products precisely when it has a terminal object and binary products.

A diagram $D : \mathcal{I} \to \mathcal{C}$ is *small* when $\mathcal{I}$ is a small category. A *small limit* is a limit of a small diagram. A *finite limit* is a limit of a diagram whose index category is finite.

Exercise A.6.6. Prove that a limit, when it exists, is unique up to isomorphism.

The following proposition and its proof tell us how to compute arbitrary limits from simpler ones. We omit detailed proofs as they can be found in any standard textbook on category theory.

Proposition A.6.7. *The following are equivalent for a category $\mathcal{C}$:*

1. *$\mathcal{C}$ has a terminal object and all pullbacks.*
2. *$\mathcal{C}$ has equalizers and all finite products.*
3. *$\mathcal{C}$ has all finite limits.*

Proof. We only show how to get binary products from pullbacks and a terminal object. For objects A and B, let P be the pullback of $!_A$ and $!_B$:

$$\begin{tikzcd}
P \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "!_B"] \\
A \arrow[r, "!_A"'] & 1
\end{tikzcd}$$

Then (P, π_0, π_1) is a product of A and B because, for all $f : X \to A$ and $g : X \to B$, it is trivially the case that $!_A \circ f = !_B \circ g$. □

Proposition A.6.8. *The following are equivalent for a category $\mathcal{C}$:*

1. *$\mathcal{C}$ has equalizers and all small products.*

2. *$\mathcal{C}$ has all small limits.*

Proof. We indicate how to construct an arbitrary limit from a product and an equalizer. Let $D : \mathcal{I} \to \mathcal{C}$ be a small diagram of an arbitrary shape $\mathcal{I}$. First form an $\mathcal{I}_0$-indexed product P and an $\mathcal{I}_1$-indexed product Q

$$P = \prod_{i \in \mathcal{I}_0} D_i \;, \qquad\qquad Q = \prod_{u \in \mathcal{I}_1} D_{\mathsf{cod}\, u} \;.$$

By the universal property of products, there are unique morphisms $f : P \to Q$ and $g : P \to Q$ such that, for all morphisms $u \in \mathcal{I}_1$,

$$\pi^Q_u \circ f = Du \circ \pi^P_{\mathsf{dom}\, u} \;, \qquad\qquad \pi^Q_u \circ g = \pi^P_{\mathsf{cod}\, u} \;.$$

Let E be the equalizer of f and g,

$$\begin{tikzcd} E \arrow[r, "e"] & P \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Q \end{tikzcd}$$

For every $i \in \mathcal{I}$ there is a morphism $\varepsilon_i : E \to D_i$, namely $\varepsilon_i = \pi^P_i \circ e$. We claim that (E, ε) is a limit of D. First, (E, ε) is a cone on D because, for all $u : i \to j$ in $\mathcal{I}$,

$$Du \circ \varepsilon_i = Du \circ \pi^P_i \circ e = \pi^Q_u \circ f \circ e = \pi^Q_u \circ g \circ e = \pi^P_j \circ e = \varepsilon_j \;.$$

If (A, α) is any cone on D there exists a unique $t : A \to P$ such that $\alpha_i = \pi^P_i \circ t$ for all $i \in \mathcal{I}$. For every $u : i \to j$ in $\mathcal{I}$ we have

$$\pi^Q_u \circ g \circ t = \pi^P_j \circ t = t_j = Du \circ t_i = Du \circ \pi^P_i \circ t = \pi^Q_u \circ f \circ t \;,$$

therefore $g \circ t = f \circ t$. This implies that there is a unique factorization $k : A \to E$ such that $t = e \circ k$. Now for every $i \in \mathcal{I}$

$$\varepsilon_i \circ k = \pi^P_i \circ e \circ k = \pi^P_i \circ t = \alpha_i$$

so that $k : A \to E$ is the required factorization of the cone (A, α) through the cone (E, ε). To see that k is unique, suppose $m : A \to E$ is another factorization such that $\alpha_i = \varepsilon_i \circ m$ for all $i \in \mathcal{I}$. Since e is mono it suffices to show that $e \circ m = e \circ k$, which is equivalent to proving $\pi^P_i \circ e \circ m = \pi^P_i \circ e \circ k$ for all $i \in \mathcal{I}$. This last equality holds because

$$\pi^P_i \circ e \circ k = \pi^P_i \circ t = \alpha_i = \varepsilon_i \circ m = \pi^P_i \circ e \circ m \;.$$

□

A category is *(small) complete* when it has all small limits, and it is *finitely complete* (or *left exact*, briefly *lex*) when it has finite limits.

Limits of presheaves Let $\mathcal{C}$ be a locally small category. Then the presheaf category $\widehat{\mathcal{C}} = \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$ has all small limits and they are computed pointwise, e.g., $(P \times Q)A = PA \times QA$ for $P, Q \in \widehat{\mathcal{C}}$, $A \in \mathcal{C}$. To see that this is really so, let $\mathcal{I}$ be a small index category and $D : \mathcal{I} \to \widehat{\mathcal{C}}$ a diagram of presheaves. Then for every $A \in \mathcal{C}$ the diagram D can be instantiated at A to give a diagram $DA : \mathcal{I} \to \mathsf{Set}$, $(DA)_i = D_i A$. Because Set is small complete, we can define a presheaf L by computing the limit of DA:

$$LA = \lim DA = \varprojlim_{i \in \mathcal{I}} D_i A \; .$$

We should keep in mind that $\lim DA$ is actually given by an object $(\lim DA)$ and a natural transformation $\delta A : \Delta_{(\lim DA)} \Longrightarrow DA$. The value of LA is supposed to be just the object part of $\lim DA$. From a morphism $f : A \to B$ we obtain for each $i \in \mathcal{I}$ a function $D_i f \circ (\delta A)_i : LA \to D_i B$, and thus a cone $(LA, Df \circ \delta A)$ on DB. Presheaf L maps the morphism $f : A \to B$ to the unique factorization $Lf : LA \Longrightarrow LB$ of the cone $(LA, Df \circ \delta A)$ on DB through the limit cone LB on DB.

For every $i \in \mathcal{I}$, there is a function $\Lambda_i = (\delta A)_i : LA \to D_i A$. The family $\{\Lambda_i\}_{i \in \mathcal{I}}$ is a natural transformation from Δ_{LA} to DA. This gives us a cone (L, Λ) on D, which is in fact a limit cone. Indeed, if (S, Σ) is another cone on D then for every $A \in \mathcal{C}$ there exists a unique function $\phi_A : SA \to LA$ because SA is a cone on DA and LA is a limit cone on DA. The family $\{\phi_A\}_{A \in \mathcal{C}}$ is the unique natural transformation $\phi : S \Longrightarrow L$ for which $\Sigma = \phi \circ \Lambda$.

A.6.6 Colimits

Colimits are the dual notion of limits. Thus, a *colimit* of a diagram $D : \mathcal{I} \to \mathcal{C}$ is a limit of the dual diagram $D^{\mathsf{op}} : \mathcal{I}^{\mathsf{op}} \to \mathcal{C}^{\mathsf{op}}$ in the dual (i.e., opposite) category $\mathcal{C}^{\mathsf{op}}$:

$$\operatorname{colim}(D : \mathcal{I} \to \mathcal{C}) = \lim(D^{\mathsf{op}} : \mathcal{I}^{\mathsf{op}} \to \mathcal{C}^{\mathsf{op}}) \; .$$

Explicitly, the colimit of a diagram $D : \mathcal{I} \to \mathcal{C}$ is the initial object in the category of *cocones* $\mathsf{Cocone}(D)$ on D. A cocone (A, α) on D is a natural transformation $\alpha : D \Longrightarrow \Delta_A$. It is given by an object $A \in \mathcal{C}$ and, for each $i \in \mathcal{I}$, a morphism $\alpha_i : D_i \to A$, such that $\alpha_i = \alpha_j \circ Du$ whenever $u : i \to j$ in $\mathcal{I}$. A morphism between cocones $f : (A, \alpha) \to (B, \beta)$ is a morphism $f : A \to B$ in $\mathcal{C}$ such that $\beta_i = f \circ \alpha_i$ for all $i \in \mathcal{I}$.

A colimit of $D : \mathcal{I} \to \mathcal{C}$ is then given by a cocone (C, ζ) on D such that, for every cocone (A, α) on D there exists a unique morphism $f : C \to A$ such that $\alpha_i = f \circ \zeta_i$ for all $i \in D$. We denote a colimit of D by one of the following:

$$\operatorname{colim} D \qquad\qquad \operatorname{colim}_{i \in \mathcal{I}} D_i \qquad\qquad \varinjlim_{i \in \mathcal{I}} D_i \; .$$

Colimits are also called *inductive limits*.

Exercise A.6.9. Formulate the dual of Proposition A.6.7 and Proposition A.6.8 for colimits (coequalizers are defined in Section A.6.9).

A.6.7 Binary coproducts

In a category $\mathcal{C}$, the *(binary) coproduct* of objects A and B is an object $A+B$ together with *injections* $\iota_0 : A \to A+B$ and $\iota_1 : B \to A+B$ such that, for every object $C \in \mathcal{C}$ and all morphisms $f : A \to C$, $g : B \to C$ there exists a *unique* morphism $h : A+B \to C$ for which the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "\iota_0"] \arrow[dr, "f"'] & A+B \arrow[d, dotted, "h"] & B \arrow[l, "\iota_1"'] \arrow[dl, "g"] \\
& C &
\end{tikzcd}$$

The arrow $h : A+B \to C$ is denoted by $[f,g]$.

The coproduct $A+B$ is the colimit of the diagram $D : \mathsf{2} \to \mathcal{C}$, where $\mathcal{I}$ is the discrete category on two objects 0 and 1, and $D_0 = A$, $D_1 = B$.

In Set the coproduct is the disjoint union, defined by

$$X+Y = \left\{\langle 0,x\rangle \;\middle|\; x \in X\right\} \cup \left\{\langle 1,y\rangle \;\middle|\; x \in Y\right\} ,$$

where 0 and 1 are distinct sets, for example $\emptyset$ and $\{\emptyset\}$. Given functions $f : X \to Z$ and $g : Y \to Z$, the unique function $[f,g] : X+Y \to Z$ is the usual *definition by cases*:

$$[f,g]u = \begin{cases} fx & \text{if } u = \langle 0,x\rangle \\ gx & \text{if } u = \langle 1,x\rangle \ . \end{cases}$$

Exercise A.6.10. Show that the categories of posets and of topological spaces both have coproducts.

A.6.8 Initial objects

An *initial object* in a category $\mathcal{C}$ is an object $\mathsf{0} \in \mathcal{C}$ such that for every $A \in \mathcal{C}$ there exists a *unique* morphism $\mathsf{o}_A : \mathsf{0} \to A$.

An initial object is the colimit of the empty diagram.

In Set, the initial object is the empty set.

Exercise A.6.11. What is the initial and what is the terminal object in the category of groups?

A *zero object* is an object that is both initial and terminal.

Exercise A.6.12. Show that in the category of Abelian[8] groups finite products and coproducts agree, that is $\mathsf{0} \cong \mathsf{1}$ and $A \times B \cong A+B$.

Exercise A.6.13. Suppose A and B are Abelian groups. Is there a difference between their coproduct in the category Group of groups, and their coproduct in the category $\mathsf{AbGroup}$ of Abelian groups?

[8] An Abelian group is one that satisfies the commutative law $x \cdot y = y \cdot x$.

A.6.9 Coequalizers

Given objects and morphisms

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "q"] & Q
\end{tikzcd}$$

we say that q *coequalizes* f and g when $e \circ f = e \circ g$. A *coequalizer* of f and g is a *universal* coequalizing morphism; thus $q : B \to Q$ is a coequalizer of f and g when it coequalizes them and, for all $s : B \to S$, if $s \circ f = s \circ g$ then there exists a *unique* morphism $r : Q \to S$ such that $s = r \circ q$:

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "q"] \arrow[dr, "s"'] & Q \arrow[d, dotted, "r"] \\
& & S
\end{tikzcd}$$

In Set the coequalizer of parallel functions $f : A \to B$ and $g : A \to B$ is the quotient set $Q = B/{\sim}$ where $\sim$ is the least equivalence relation on B satisfying

$$fx = gy \Rightarrow x \sim y \; .$$

The function $q : B \to Q$ is the canonical quotient map which assigns to each element $x \in B$ its equivalence class $[x] \in B/{\sim}$. In general, a coequalizer can be thought of as the quotient by the equivalence relation generated by the corresponding equation.

Exercise A.6.14. Show that a coequalizer is an epimorphism, i.e., if $q : B \to Q$ is a coequalizer of f and g, then, for all $u, v : Q \to T$, $u \circ q = v \circ q$ implies $u = v$. [Hint: use the duality between limits and colimits and Exercise A.6.3.]

Definition A.6.15. A morphism is a *regular epi* if it is a coequalizer.

The difference between epis and regular epis is also illustrated in the category Top: a continuous map $f : X \to Y$ is epi when it is surjective, whereas it is a regular epi when it is a topological quotient map.[9]

A.6.10 Pushouts

A *pushout* of $f : A \to B$ and $g : A \to C$ is an object Q with morphisms $q_0 : B \to Q$ and $q_1 : C \to Q$ such that $q_0 \circ f = q_1 \circ g$, and whenever $r_0 : B \to R$, $r_1 : C \to R$ are such that

[9] A continuous map $f : X \to Y$ is a topological quotient map when it is surjective and, for every $U \subseteq Y$, U is open if, and only if, f^*U is open.

$r_0 \circ f = r_1 \circ g$, then there exists a unique $s : Q \to R$ such that $r_0 = s \circ q_0$ and $r_1 = s \circ q_1$:

$$\begin{tikzcd}
A \arrow[r, "g"] \arrow[d, "f"'] & C \arrow[d, "q_1"] \arrow[ddr, bend left, "r_1"] & \\
B \arrow[r, "q_0"'] \arrow[drr, bend right, "r_0"'] & Q \arrow[dr, dotted, "s"] \arrow[ul, phantom, "\ulcorner", very near start] & \\
& & R
\end{tikzcd}$$

We indicate that Q is a pushout by drawing a square corner next to it, as in the above diagram. The above pushout Q is sometimes denoted by $B +_A C$.

A pushout as above is a colimit of the diagram $D : \mathcal{I} \to \mathcal{C}$ where the index category $\mathcal{I}$ is

$$\begin{tikzcd}
\bullet \arrow[r, "2"] \arrow[d, "1"'] & \bullet \\
\bullet &
\end{tikzcd}$$

and $D1 = f$, $D2 = g$.

In Set, the pushout of $f : A \to C$ and $g : B \to C$ is the quotient set

$$Q = (B + C)/\!\sim$$

where $B + C$ is the disjoint union of B and C, and $\sim$ is the least equivalence relation on $B + C$ such that, for all $x \in A$,

$$fx \sim gx \;.$$

The functions $q_0 : B \to Q$, $q_1 : C \to Q$ are the injections, $q_0 x = [x]$, $q_1 y = [y]$, where $[x]$ is the equivalence class of x.

A.6.11 Limits as adjoints

Limits and colimits can be defined as adjoints to certain very simple functors.

First, observe that an object $A \in \mathcal{C}$ can be viewed as a functor from the terminal category $\mathsf{1}$ to $\mathcal{C}$, namely the functor which maps the only object $\star$ of $\mathsf{1}$ to A. Since $\mathsf{1}$ is the terminal object in Cat, there exists a unique functor $!_{\mathcal{C}} : \mathcal{C} \to \mathsf{1}$, which maps every object of $\mathcal{C}$ to $\star$.

Now we can ask whether this simple functor $!_{\mathcal{C}} : \mathcal{C} \to \mathsf{1}$ has any adjoints. Indeed, it has a right adjoint just if $\mathcal{C}$ has a terminal object $1_{\mathcal{C}}$, for the corresponding functor $1_{\mathcal{C}} : \mathsf{1} \to \mathcal{C}$ has the property that, for every $A \in \mathcal{C}$ we have a (trivially natural) bijective correspondence:

$$\begin{array}{c} !_A : A \to 1_{\mathcal{C}} \\ \hline\hline 1_\star : !_{\mathcal{C}} A \to \star \end{array}$$

Similarly, an initial object is a left adjoint to $!_{\mathcal{C}}$:

$$\mathsf{0}_{\mathcal{C}} \dashv !_{\mathcal{C}} \dashv \mathsf{1}_{\mathcal{C}} \ .$$

Now consider the diagonal functor,

$$\Delta : \mathcal{C} \to \mathcal{C} \times \mathcal{C},$$

defined by $\Delta A = \langle A, A \rangle$, $\Delta f = \langle f, f \rangle$. When does this have adjoints?

If $\mathcal{C}$ has all binary products, then they determine a functor

$$- \times - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$$

which maps $\langle A, B \rangle$ to $A \times B$ and a pair of morphisms $\langle f : A \to A', g : B \to B' \rangle$ to the unique morphism $f \times g : A \times B \to A' \times B'$ for which $\pi_0 \circ (f \times g) = f \circ \pi_0$ and $\pi_1 \circ (f \times g) = g \circ \pi_1$,

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \times B \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] \arrow[d, "f \times g"] & B \arrow[d, "g"] \\
A' & A' \times B' \arrow[l, "\pi_0"] \arrow[r, "\pi_1"'] & B'
\end{tikzcd}$$

The product functor $\times$ is right adjoint to the diagonal functor Δ. Indeed, there is a natural bijective correspondence:

$$\frac{\langle f, g \rangle : \langle A, A \rangle \to \langle B, C \rangle}{\overline{f \times g : A \to B \times C}}$$

Similarly, binary coproducts are easily seen to be left adjoint to the diagonal functor,

$$+ \dashv \Delta \dashv \times \ .$$

Now in general, consider limits of shape $\mathcal{I}$ in a category $\mathcal{C}$. There is the constant diagram functor

$$\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{I}}$$

that maps $A \in \mathcal{C}$ to the constant diagram $\Delta_A : \mathcal{I} \to \mathcal{C}$. The limit construction is a functor

$$\varprojlim : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$$

that maps each diagram $D \in \mathcal{C}^{\mathcal{I}}$ to its limit $\varprojlim D$. These two are adjoint, $\Delta \dashv \varprojlim$, because there is a natural bijective correspondence between cones $\alpha : \Delta_A \Longrightarrow D$ on D, and their factorizations through the limit of D,

$$\frac{\alpha : \Delta_A \Longrightarrow D}{\overline{A \to \varprojlim D}}$$

An analogous correspondence holds for colimits, so that we obtain a pair of adjunctions,

$$\varinjlim \dashv \Delta \dashv \varprojlim \ ,$$

which, of course, subsume all the previously mentioned cases.

Exercise A.6.16. How are the functors $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{I}}$, $\varinjlim : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$, and $\varprojlim : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$ defined on morphisms?

A.6.12 Preservation of limits

We say that a functor $F : \mathcal{C} \to \mathcal{D}$ *preserves products* when, given a product

$$\begin{tikzcd} A & A \times B \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] & B \end{tikzcd}$$

its image in $\mathcal{D}$,

$$\begin{tikzcd} FA & F(A \times B) \arrow[l, "F\pi_0"'] \arrow[r, "F\pi_1"] & FB \end{tikzcd}$$

is a product of FA and FB. If $\mathcal{D}$ has chosen binary products, F preserves binary products if, and only if, the unique morphism $f : F(A \times B) \to FA \times FB$ which makes the following diagram commutative is an isomorphism: [10]

$$\begin{tikzcd} & F(A \times B) \arrow[dl, "F\pi_0"'] \arrow[d, "f"] \arrow[dr, "F\pi_1"] & \\ FA & FA \times FB \arrow[l, "\pi_0"] \arrow[r, "\pi_1"'] & FB \end{tikzcd}$$

In general, a functor $F : \mathcal{C} \to \mathcal{D}$ is said to *preserve limits* of shape $\mathcal{I}$ when it maps limit cones to limit cones: if (L, λ) is a limit of $D : \mathcal{I} \to \mathcal{C}$ then $(FL, F \circ \lambda)$ is a limit of $F \circ D : \mathcal{I} \to \mathcal{D}$.

Analogously, a functor $F : \mathcal{C} \to \mathcal{D}$ is said to *preserve colimits* of shape $\mathcal{I}$ when it maps colimit cocones to colimit cocones: if (C, ζ) is a colimit of $D : \mathcal{I} \to \mathcal{C}$ then $(FC, F \circ \zeta)$ is a colimit of $F \circ D : \mathcal{I} \to \mathcal{D}$.

Proposition A.6.17. *(a) A functor preserves finite (small) limits if, and only if, it preserves equalizers and finite (small) products. (b) A functor preserves finite (small) colimits if, and only if, it preserves coequalizers and finite (small) coproducts.*

Proof. This follows from the fact that limits are constructed from equalizers and products, cf. Proposition A.6.8, and that colimits are constructed from coequalizers and coproducts, cf. Exercise A.6.9. □

Proposition A.6.18. *For a locally small category $\mathcal{C}$, the Yoneda embedding* $\mathsf{y} : \mathcal{C} \to \widehat{\mathcal{C}}$ *preserves all limits that exist in $\mathcal{C}$.*

[10] Products are determined up to isomorphism only, so it would be too restrictive to require $F(A \times B) = FA \times FB$. When that is the case, however, we say that the functor F *strictly* preserves products.

Proof. Suppose (L, λ) is a limit of $D : \mathcal{I} \to \mathcal{C}$. The Yoneda embedding maps D to the diagram $\mathsf{y} \circ D : \mathcal{I} \to \widehat{\mathcal{C}}$, defined by

$$(\mathsf{y} \circ D)_i = \mathsf{y} D_i = \mathcal{C}(-, D_i) \; .$$

and it maps the limit cone (L, λ) to the cone $(\mathsf{y}L, \mathsf{y} \circ \lambda)$ on $\mathsf{y} \circ D$, defined by

$$(\mathsf{y} \circ \lambda)_i = \mathsf{y}\lambda_i = \mathcal{C}(-, \lambda_i) \; .$$

To see that $(\mathsf{y}L, \mathsf{y} \circ \lambda)$ is a limit cone on $\mathsf{y} \circ D$, consider a cone (M, μ) on $\mathsf{y} \circ D$. Then $\mu : \Delta_M \Longrightarrow D$ consists of a family of functions, one for each $i \in \mathcal{I}$ and $A \in \mathcal{C}$,

$$(\mu_i)_A : MA \to \mathcal{C}(A, D_i) \; .$$

For every $A \in \mathcal{C}$ and $m \in MA$ we get a cone on D consisting of morphisms

$$(\mu_i)_A m : A \to D_i \; . \qquad (i \in \mathcal{I})$$

There exists a unique morphism $\phi_A m : A \to L$ such that $(\mu_i)_A m = \lambda_i \circ \phi_A m$. The family of functions

$$\phi_A : MA \to \mathcal{C}(A, L) = (\mathsf{y} \circ L)A \qquad (A \in \mathcal{C})$$

forms a factorization $\phi : M \Longrightarrow \mathsf{y}L$ of the cone (M, μ) through the cone (L, λ). This factorization is unique because each $\phi_A m$ is unique. □

In effect we showed that a covariant representable functor $\mathcal{C}(A, -) : \mathcal{C} \to \mathsf{Set}$ preserves existing limits,

$$\mathcal{C}(A, \varprojlim_{i \in \mathcal{I}} D_i) \cong \varprojlim_{i \in \mathcal{I}} \mathcal{C}(A, D_i) \; .$$

By duality, the contravariant representable functor $\mathcal{C}(-, A) : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$ maps existing colimits to limits,

$$\mathcal{C}(\varinjlim_{i \in \mathcal{I}} D_i, A) \cong \varprojlim_{i \in \mathcal{I}} \mathcal{C}(D_i, A) \; .$$

Exercise A.6.19. Prove the above claim that a contravariant representable functor $\mathcal{C}(-, A) : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$ maps existing colimits to limits. Use duality between limits and colimits. Does it also follow *by a simple duality argument* that a contravariant representable functor $\mathcal{C}(-, A)$ maps existing limits to colimits? How about a covariant representable functor $\mathcal{C}(A, -)$ mapping existing colimits to limits?

Exercise A.6.20. Prove that a functor $F : \mathcal{C} \to \mathcal{D}$ preserves monos if it preserves limits. In particular, the Yoneda embedding preserves monos. Hint: Exercise A.6.5.

Proposition A.6.21. *Right adjoints preserve limits, and left adjoints preserve colimits.*

Proof. Suppose we have adjoint functors

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

and a diagram $D : \mathcal{I} \to \mathcal{D}$ whose limit exists in $\mathcal{D}$. We would like to use the following slick application of Yoneda Lemma to show that G preserves limits: for every $A \in \mathcal{C}$,

$$\begin{aligned} \mathcal{C}(A, G(\varprojlim D)) \cong \mathcal{D}(FA, \varprojlim D) \cong \varprojlim_{i \in \mathcal{I}} \mathcal{D}(FA, D_i) \\ \cong \varprojlim_{i \in \mathcal{I}} \mathcal{C}(A, GD_i) \cong \mathcal{C}(A, \varprojlim (G \circ D)) \ . \end{aligned}$$

Therefore $G(\lim D) \cong \lim(G \circ D)$. However, this argument only works if we already know that the limit of $G \circ D$ exists.

We can also prove the stronger claim that whenever the limit of $D : \mathcal{I} \to \mathcal{D}$ exists then the limit of $G \circ D$ exists in $\mathcal{C}$ and its limit is $G(\lim D)$. So suppose (L, λ) is a limit cone of D. Then $(GL, G \circ \lambda)$ is a cone on $G \circ D$. If (A, α) is another cone on $G \circ D$, we have by adjunction a cone (FA, γ) on D,

$$\frac{\alpha_i : A \to GD_i}{\gamma_i : FA \to D_i}$$

There exists a unique factorization $f : FA \to L$ of this cone through (L, λ). Again by adjunction, we obtain a unique factorization $g : A \to GL$ of the cone (A, α) through the cone $(GL, G \circ \lambda)$:

$$\frac{f : FA \to L}{g : A \to GL}$$

The factorization g is unique because γ is uniquely determined from α, f uniquely from α, and g uniquely from f.

By a dual argument, a left adjoint preserves colimits. □

Appendix B

Logic

B.1 Concrete and abstract syntax

By *syntax* we generally mean manipulation of finite strings of symbols according to given *grammatical rules*. For instance, the strings "7)6 + /(8" and "$(6 + 8)/7$" both consist of the same symbols but you will recognize one as junk and the other as *well formed* because you have (implicitly) applied the grammatical rules for arithmetical expressions.

Grammatical rules are usually quite complicated, as they need to prescribe associativity of operators (does "$5+6+7$" mean "$(5+6)+7$" or "$5+(6+7)$"?) and their precedence (does "$6+8/7$" mean "$(6+8)/7$" or "$6+(8/7)$"?), the role of *white space* (empty space between symbols and line breaks), rules for nesting and balancing parentheses, etc. It is not our intention to dwell on such details, but rather to focus on the mathematical nature of well-formed expressions, namely that they represent inductively generated finite trees.[1] Under this view the string "$(6+8)/7$" is just a concrete representation of the tree depicted in Figure B.1.

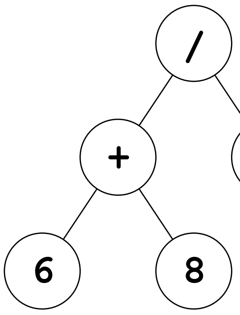

Figure B.1: The tree represented by $(6 + 8)/7$

Concrete representation of expressions as finite strings of symbols is called *concrete syntax*, while in *abstract syntax* we view expressions as finite trees. The passage from the

[1] We are limiting attention to the so-called *context-free* grammar, which are sufficient for our purposes. More complicated grammars are rarely used to describe formal languages in logic and computer science.

former to the latter is called *parsing* and is beyond the scope of this book. We will always specify only abstract syntax and assume that the corresponding concrete syntax follows the customary rules for parentheses, associativity and precedence of operators.

As an illustration we give rules for the (abstract) syntax of propositional calculus in *Backus-Naur* form:

$$\text{Propositional variable } p ::= \mathtt{p}_1 \mid \mathtt{p}_2 \mid \mathtt{p}_3 \mid \cdots$$
$$\text{Propositional formula } \phi ::= p \mid \bot \mid \top \mid \phi_1 \land \phi_2 \mid \phi_1 \lor \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \neg\phi$$

The vertical bars should be read as "or". The first rule says that a propositional variable is the constant $\mathtt{p}_1$, or the constant $\mathtt{p}_2$, or the constant $\mathtt{p}_3$, etc.[2] The second rule tells us that there are seven inductive rules for building a propositional formula:

- a propositional variable is a formula,
- the constants $\bot$ and $\top$ are formulas,
- if ϕ_1, ϕ_2, and ϕ are formulas, then so are $\phi_1 \land \phi_2$, $\phi_1 \lor \phi_2$, $\phi_1 \Rightarrow \phi_2$, and $\neg\phi$.

Even though abstract syntax rules say nothing about parentheses or operator associativity and precedence, we shall rely on established conventions for mathematical notation and write down concrete representations of propositional formulas, e.g., $\mathtt{p}_4 \land (\mathtt{p}_1 \lor \neg \mathtt{p}_1) \land \mathtt{p}_4 \lor \mathtt{p}_2$.

A word of warning: operator associativity in syntax is not to be confused with the usual notion of associativity in mathematics. We say that an operator $\star$ is *left associative* when an expression $x \star y \star z$ represents the left-hand tree in Figure B.2, and *right associative* when it represents the right-hand tree. Thus the usual operation of subtraction $-$ is left associative, but is not associative in the usual mathematical sense.

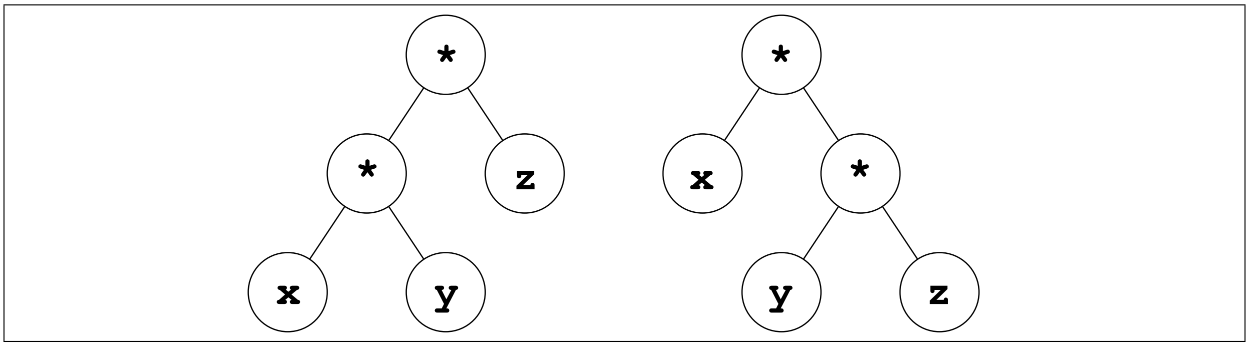

Figure B.2: Left and right associativity of $x \star y \star z$

[2] In an actual computer implementation we would allow arbitrary finite strings of letters as propositional variables. In logic we only care about the fact that we can never run out of fresh variables, i.e., that there are countably infinitely many of them.

B.2 Free and bound variables

Variables appearing in an expression may be *free* or *bound*. For example, in expressions

$$\int_0^1 \sin(a \cdot x)\, dx, \qquad x \mapsto ax^2 + bx + c, \qquad \forall\, x\,.\,(x < a \lor x > b)$$

the variables a, b and c are free, while x is bound by the integral operator $\int$, the function formation $\mapsto$, and the universal quantifier $\forall$, respectively. To be quite precise, it is an *occurrence* of a variable that is free or bound. For example, in expression $\phi(x) \lor \exists\, x\,.\,A\psi(x, x)$ the first occurrence of x is free and the remaining ones are bound.

In this book the following operators bind variables:

- quantifiers $\exists$ and $\forall$, cf. **??**,
- λ-abstraction, cf. **??**,
- search for others **??**.

When a variable is bound we may always rename it, provided the renaming does not confuse it with another variable. In the integral above we could rename x to y, but not to a because the binding operation would *capture* the free variable a to produce the unintended $\int_0^1 \sin(a^2)\, da$. Renaming of bound variables is called *α-renaming.*

We consider two expressions *equal* if they only differ in the names of bound variables, i.e., if one can be obtained from the other by α-renaming. Furthermore, we adhere to *Barendregt's variable convention [**?**, p. 2]*, which says that bound variables are always chosen so as to differ from free variables. Thus we would never write $\phi(x) \lor \exists\, x\,.\,A\psi(x, x)$ but rather $\phi(x) \lor \exists\, y\,.\,A\psi(y, y)$. By doing so we need not worry about capturing or otherwise confusing free and bound variables.

In logic we need to be more careful about variables than is customary in traditional mathematics. Specifically, we always specify which free variables may appear in an expression.[3] We write

$$x_1 : A_1, \ldots, x_n : A_n \mid t$$

to indicate that expression t may contain only free variables $x_1, \ldots, x_n$ of types $A_1, \ldots, A_n$. The list

$$x_1 : A_1, \ldots, x_n : A_n$$

is called a *context* in which t appears. To see why this is important consider the different meaning that the expression $x^2 + y^2 \leq 1$ recevieves in different contexts:

- $x : \mathbb{Z}, y : \mathbb{Z} \mid x^2 + y^2 \leq 1$ denotes the set of tuples $\{(-1, 0), (0, 1), (1, 0), (0, -1)\}$,
- $x : \mathbb{R}, y : \mathbb{R} \mid x^2 + y^2 \leq 1$ denotes the closed unit disc in the plane, and

[3] This is akin to one of the guiding principles of good programming language design, namely, that all variables should be *declared* before they are used.

- $x : \mathbb{R}, y : \mathbb{R}, z : \mathbb{R} \mid x^2 + y^2 \leq 1$ denotes the infinite cylinder in space whose base is the closed unit disc.

In single-sorted theories there is only one type or sort A. In this case we abbreviate a context by listing just the variables, $x_1, \ldots, x_n$.

B.3 Substitution

Substitution is a basic syntactic operation which replaces (free occurrences of) distinct variables $x_1, \ldots, x_n$ in an expression t with expressions $t_1, \ldots, t_n$, which is written as

$$t[t_1/x_1, \ldots, t_n/x_n].$$

We sometimes abbreviate this as $t[\vec{t}/\vec{x}]$ where $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{t} = (t_1, \ldots, t_n)$. Here are several examples:

$$\begin{aligned}(x^2 + x + y)[(2+3)/x] &= (2+3)^2 + (2+3) + y \\ (x^2 + y)[y/x, x/y] &= y^2 + x \\ \big(\forall x \,.\, \big(x^2 < y + x^3\big)\big)\,[x + y/y] &= \forall z \,.\, \big(z^2 < (x+y) + z^3\big)\,.\end{aligned}$$

Notice that in the third example we first renamed the bound variable x to z in order to avoid a capture by $\forall$.

Substitution is simple to explain in terms of trees. Assuming Barendregt's convention, the substitution $t[u/x]$ means that in the tree t we replace the leaves labeled x by copies of the tree u. Thus a substitution never changes the structure of the tree–it only "grows" new subtrees in places where the substituted variables occur as leaves.

Substitution satisfies the distributive law

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

provided x and y are distinct variables. There is also a corresponding multivariate version which is written the same way with a slight abuse of vector notation:

$$(t[\vec{u}/\vec{x}])[\vec{v}/\vec{y}] = (t[\vec{v}/\vec{y}])[\vec{u}[\vec{v}/\vec{y}]/\vec{x}].$$

B.4 Judgments and deductive systems

A formal system, such as first-order logic or type theory, concerns itself with *judgments*. There are many kinds of judgments, such as:

- The most common judgments are equations and other logical statements. We distinguish a formula ϕ and the judgment "ϕ holds" by writing the latter as

  $$\vdash \phi\,.$$

  The symbol $\vdash$ is generally used to indicate judgments.

- Typing judgments
$$\vdash t : A$$
expressing the fact that a term t has type A. This is not to be confused with the set-theoretic statement $t \in u$ which says that individuals t and u (of type "set") are in relation "element of" $\in$.

- Judgments expressing the fact that a certain entity is well formed. A typical example is a judgment
$$\vdash x_1 : A_1, \ldots, x_n : A_n \quad \texttt{ctx}$$
which states that $x_1 : A_1, \ldots, x_n : A_n$ is a well-formed context. This means that $x_1, \ldots, x_n$ are distinct variables and that $A_1, \ldots, A_n$ are well-formed types. This kind of judgement is often omitted and it is tacitly assumed that whatever entities we deal with are in fact well-formed.

A *hypothetical judgement* has the form

$$H_1, \ldots, H_n \vdash C$$

and means that hypotheses $H_1, \ldots, H_n$ entail consequence C (with respect to a given decuctive system). We may also add a typing context to get a general form of judgment

$$x_1 : A_1, \ldots, x_n : A_n \mid H_1, \ldots, H_m \vdash C.$$

This should be read as: "if $x_1, \ldots, x_n$ are variables of types $A_1, \ldots, A_n$, respectively, then hypotheses $H_1, \ldots, H_m$ entail conclusion C." For our purposes such contexts will suffice, but you should not be surprised to see other kinds of judgments in logic.

A *deductive system* is a set of inference rules for deriving judgments. A typical inference rule has the form

$$\frac{J_1 \quad J_2 \quad \cdots \quad J_n}{J}\ C$$

This means that we can infer judgment J if we have already derived judgments $J_1, \ldots, J_n$, provided that the optional side-condition C is satisfied. An *axiom* is an inference rule of the form

$$\frac{}{J}$$

A *two-way rule*

$$\begin{array}{c} J_1 \quad J_2 \quad \cdots \quad J_n \\ \hline\hline K_1 \quad K_2 \quad \cdots \quad K_m \end{array}$$

is a combination of $n+m$ inference rules stating that we may infer each K_i from $J_1, \ldots, J_n$ and each J_i from $K_1, \ldots, K_m$.

A *derivation* of a judgment J is a finite tree whose root is J, the nodes are inference rules, and the leaves are axioms. An example is presented in the next subsection.

The set of all judgments that hold in a given deductive system is generated inductively by starting with the axioms and applying inference rules.

B.5 Example: Equational reasoning

Equational reasoning is so straightforward that one almost doesn't notice it, consisting mainly, as it does, of "substituting equals for equals". The only judgements are equations between terms, $s = t$, which consist of function symbols, constants, and variables. The inference rules are just the usual ones making $s = t$ a congruence relation on the terms. More formally, we have the following specification of what may be called the *equational calculus*.

$$\begin{aligned}
\text{Variable } v &::= x \mid y \mid z \mid \cdots \\
\text{Constant symbol } c &::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\
\text{Function symbol } f^k &::= \mathtt{f}_1^{k_1} \mid \mathtt{f}_2^{k_2} \mid \cdots \\
\text{Term } t &::= v \mid c \mid f^k(t_1, \ldots, t_k)
\end{aligned}$$

The superscript on the function symbol f^k indicates the arity.

The equational calculus has just one form of judgement

$$x_1, \ldots, x_n \mid t_1 = t_2$$

where $x_1, \ldots, x_n$ is a *context* consisting of distinct variables, and the variables in the equation must occur among the ones listed in the context.

There are four inference rules for the equational calculus. They may be assumed to leave the contexts unchanged, which may therefore be omitted.

$$\frac{}{t = t} \qquad \frac{t_1 = t_2}{t_2 = t_1} \qquad \frac{t_1 = t_2,\ t_2 = t_3}{t_1 = t_3} \qquad \frac{t_1 = t_2,\ t_3 = t_4}{t_1[t_3/x] = t_2[t_4/x]}$$

An *equational theory* $\mathbb{T}$ consists of a set of constant and function symbols (with arities), and a set of equations, called *axioms*.

B.6 Example: Predicate calculus

We spell out the details of single-sorted predicate calculus and first-order theories. This is the most common deductive system taught in classical courses on logic.

The predicate calculus has the following syntax:

$$\begin{aligned}
\text{Variable } v &::= x \mid y \mid z \mid \cdots \\
\text{Constant symbol } c &::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\
\text{Function symbol}^4\ f^k &::= \mathtt{f}_1^{k_1} \mid \mathtt{f}_2^{k_2} \mid \cdots \\
\text{Term } t &::= v \mid c \mid f^k(t_1, \ldots, t_k) \\
\text{Relation symbol } R^m &::= \mathtt{R}_1^{m_1} \mid \mathtt{R}_2^{m_2} \mid \cdots \\
\text{Formula } \phi &::= \bot \mid \top \mid R^m(t_1, \ldots, t_m) \mid t_1 = t_2 \mid \\
&\qquad \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \neg\phi \mid \forall\, x \,.\, \phi \mid \exists\, x \,.\, \phi.
\end{aligned}$$

The variable x is bound in $\forall\, x \,.\, \phi$ and $\exists\, x \,.\, \phi$.

The predicate calculus has one form of judgement

$$x_1, \ldots, x_n \mid \phi_1, \ldots, \phi_m \vdash \phi$$

where $x_1, \ldots, x_n$ is a *context* consisting of distinct variables, $\phi_1, \ldots, \phi_m$ are *hypotheses* and ϕ is the *conclusion*. The free variables in the hypotheses and the conclusion must occur among the ones listed in the context. We abbreviate the context with Γ and Φ with hypotheses. Because most rules leave the context unchanged, we omit the context unless something interesting happens with it.

The following inference rules are given in the form of adjunctions. See Appendix **??** for the more usual formulation in terms of introduction an elimination rules.

$$\overline{\phi_1, \ldots, \phi_m \vdash \phi_i} \qquad \overline{\Phi \vdash \top} \qquad \overline{\Phi, \bot \vdash \phi}$$

$$\begin{array}{c} \Phi \vdash \phi_1 \quad \Phi \vdash \phi_2 \\ \hline\hline \Phi \vdash \phi_1 \wedge \phi_2 \end{array} \qquad \begin{array}{c} \Phi, \phi_1 \vdash \psi \quad \Phi, \phi_2 \vdash \psi \\ \hline\hline \Phi, \phi_1 \vee \phi_2 \vdash \psi \end{array} \qquad \begin{array}{c} \Phi, \phi_1 \vdash \phi_2 \\ \hline\hline \Phi \vdash \phi_1 \Rightarrow \phi_2 \end{array}$$

$$\begin{array}{c} \Gamma, x, y \mid \Phi, x = y \vdash \phi \\ \hline\hline \Gamma, x \mid \Phi \vdash \phi[x/y] \end{array} \qquad \begin{array}{c} \Gamma, x \mid \Phi, \phi \vdash \psi \\ \hline\hline \Gamma \mid \Phi, \exists\, x \,.\, \phi \vdash \psi \end{array} \qquad \begin{array}{c} \Gamma, x \mid \Phi \vdash \phi \\ \hline\hline \Gamma \mid \Phi \vdash \forall\, x \,.\, \phi \end{array}$$

The equality rule implicitly requires that y does not appear in Φ, and the quantifier rules implicitly require that x does not occur freely in Φ and ψ because the judgments below the lines are supposed to be well formed.

Negation $\neg\phi$ is defined to be $\phi \Rightarrow \bot$. To obtain *classical* logic we also need the law of excluded middle,

$$\overline{\Phi \vdash \phi \vee \neg\phi}$$

Comment on the fact that contraction and weakening are admissible.

Give an example of a derivation.

A *first-order theory* $\mathbb{T}$ consists of a set of constant, function and relation symbols with corresponding arities, and a set of formulas, called *axioms*.

Give examples of a first-order theories.

Appendix C

Formalities

Pages upon pages of formal rules.

Bibliography

[ALR03] J. Adamek, F. W. Lawvere, and J. Rosiky. On the duality between varieties and algebraic theories. *Algebra Universalis*, 49:35–49, 2003.

[Awo10] Steve Awodey. *Category Theory.* Number 52 in Oxford Logic Guides. Oxford University Press, 2010.

[Bor94] F. Borceux. *Handbook of Categorical Algebra II. Categories and Structures*, volume 51 of *Encyclopedia of Mathematics and Its Applications.* Cambridge University Press, 1994.

[But98] C. Butz. Regular categories and regular logic. BRICS Lecture Series, 1998.

[Fre72] Peter Freyd. Aspects of topoi. *Bulletin of the Australian Mathematical Society*, 7:1–76, 1072.

[GER96] Houman Zolfaghari Gonzalo E. Reyes. Bi-heyting algebras, toposes and modalities. *J. Phi. Logic*, 25:25–43, 1996.

[Joh03] P.T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium, 2 vol.s.* Number 43 in Oxford Logic Guides. Oxford University Press, 2003.

[Lan71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, New York, 1971.

[Law63a] F. W. Lawvere. Functorial semantics of algebraic theories. Ph.D. thesis, Columbia University, 1963.

[Law63b] F. W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci.*, 50:869–872, 1963.

[Law65] F. W. Lawvere. Algebraic theories, algebraic categories, and algebraic functors. In *The Theory of Models*, pages 413–418. North-Holland, 1965.

[Law69] F.W. Lawvere. Adjointness in foundations. *Dialectica*, 23:281–296, 1969.

[Law70] F.W. Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. *Proceedings of the AMS Symposium on Pure Mathematics XVII*, pages 1–14, 1970.

[Law91] F. W. Lawvere. Intrinsic co-heyting boundaries and the leibniz rule in certain toposes. In G. Rosolini A. Carboni, M. Pedicchio, editor, *Category Theory - Como 1990*, number 1488 in LNM. Springer-Verlag, Heidelberg, 1991.

[McC93] W. McCune. Single axioms for groups and abelian groups with various operations. *Journal of Automated Reasoning*, 10(1):1–13, 1993.

[MM92] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory.* Springer-Verlag, New York, 1992.

[MR91] Ieke Moerdijk and Gonzalo Reyes. *Models for Smooth Infinitesimal Analysis.* Springer-Verlag, New York, 1991.

[MR95] Michael Makkai and Gonzalo Reyes. Completeness results for intuitionistic and modal logic in a categorical setting. *Annals of Pure and Applied Logic*, 72:25–101, 1995.

[PV07] E. Palmgren and S.J. Vickers. Partial horn logic and cartesian categories. *Annals of Pure and Applied Logic*, 145(3):314–353, 2007.