Introduction to Categorical Logic

[DRAFT: September 15, 2024]

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*:

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

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

Definition 1.1.2 (cf. Definition 1.2.10)**.** 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:

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

Example 1.1.4. The "empty" or trivial theory $\mathbb{T}_0$ 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 $x \leq y \Leftrightarrow x = (x \wedge y)$ 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 by considering the *category of models* of the theory, *i.e.* the category of posets, and showing that it lacks a general property enjoyed by all categories of algebras.

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:

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

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:

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

   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 1.2.10)**.** 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 $\Sigma_{\mathbb{T}}$,

$$f^I : I^k \longrightarrow I \qquad \text{in } \mathcal{C},$$

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 trivial theory $\mathbb{T}_0$ in $\mathcal{C}$ is therefore just an object A in $\mathcal{C}$, 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 by 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,

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

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 $\mathsf{eval}_C(G) = G(C)$ for each $C \in \mathbb{C}$ and group homomorphisms $h_G : G(C) \to G(D)$ for each $h : C \to D$, 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 $U \circ H$, where $U : \mathsf{Group} \to \mathsf{Set}$ is the forgetful functor, so that for each $C \in \mathbb{C}$ we have a group HC with underlying set $UHC = |HC|$. These constructions are clearly mutually inverse (up to canonical isomorphisms determined by the choice of products). Thus, briefly, *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

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

as 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, and 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 equalities were "axioms". We first describe this construction of a "syntactic category" $\mathsf{Syn}(\mathbb{T})$ for an algebraic theory $\mathbb{T}$, and then determine a universal characterization of it.

As objects of $\mathsf{Syn}(\mathbb{T})$ we take the *contexts*, i.e. sequences of distinct variables,

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

Actually, it will be more 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; but it will be useful to continue to write them as contexts.

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 formally imply that $t_k = s_k$ for every $k = 1, \ldots, n$,

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

Here we are using the usual notion of equational deduction $\mathbb{T} \vdash$ (see Section B.5). Strictly speaking, morphisms are thus *equivalence classes* of tuples 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 two morphisms

$$(t_1, \ldots, t_m) : [x_1, \ldots, x_k] \longrightarrow [x_1, \ldots, x_m]$$
$$(s_1, \ldots, s_n) : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n]$$

is the morphism $(r_1, \ldots, r_n)$ 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:

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

The identity morphism on the object $[x_1, \ldots, x_n]$ is the equivalence class of $(x_1, \ldots, x_n)$.

Using the usual rules of deduction for equational logic (Section B.5), it is easy to verify that these specifications are well-defined on equivalence classes, and therefore make $\mathsf{Syn}(\mathbb{T})$ a category.

Definition 1.1.15. The category $\mathsf{Syn}(\mathbb{T})$ just defined is called the *syntactic category* of the algebraic theory $\mathbb{T}$.

The syntactic category $\mathsf{Syn}(\mathbb{T})$ (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 $\mathsf{Syn}(\mathbb{T})$ (i.e. up to isomorphism). In this sense, the category $\mathsf{Syn}(\mathbb{T})$ is the abstract, algebraic object presented by the "generators and relations" (the operations and equations) of $\mathbb{T}$. But there is another, more important, sense in which $\mathsf{Syn}(\mathbb{T})$ represents $\mathbb{T}$, as we next show.

Exercise 1.1.16. Show that the syntactic category $\mathsf{Syn}(\mathbb{T})$ has all finite products.

1.1.3 Models as functors

Having represented an algebraic theory $\mathbb{T}$ by the syntactic category $\mathsf{Syn}(\mathbb{T})$ constructed from it, we next show that $\mathsf{Syn}(\mathbb{T})$ has the universal property that models of $\mathbb{T}$ correspond uniquely to certain functors from $\mathsf{Syn}(\mathbb{T})$. 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,

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

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

To give the details of the correspondence (1.5), let $\mathbb{T}$ be an arbitrary algebraic theory and $\mathsf{Syn}(\mathbb{T})$ the syntactic category constructed from $\mathbb{T}$ as in Definition 1.1.15. It is easy to show that the product in $\mathsf{Syn}(\mathbb{T})$ 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 $\mathsf{Syn}(\mathbb{T})$ has all finite products, including $\mathsf{1} = [-]$, the empty context (see Exercise 1.1.16). Moreover, there is a distinguished $\mathbb{T}$-model $\mathcal{U}$ in $\mathsf{Syn}(\mathbb{T})$ essentially consisting of the signature $\Sigma_{\mathbb{T}}$ itself, which we call the *syntactic model*: the underlying object $U = |\mathcal{U}|$ is the context $[x_1]$ of length one, and each operation symbol f, of say arity k, is interpreted as "itself", namely:

$$\begin{tikzcd}
U^k \arrow[r, "f^{\mathcal{U}}"] \arrow[d, "="'] & U \arrow[d, "="] \\
{[x_1, \ldots, x_k]} \arrow[r, "{[\, f(x_1, \ldots, x_k)\,]}"'] & {[x_1]}
\end{tikzcd} \tag{1.6}$$

The axioms are then satisfied, because the equivalence relation on terms is just $\mathbb{T}$-provable equality (see Section B.5). Explicitly, for all terms s, t we have:

$$\mathcal{U} \models s = t \qquad \Longleftrightarrow \qquad \mathbb{T} \vdash s = t. \tag{1.7}$$

We record this fact as the following.

Proposition 1.1.17. *The syntactic model $\mathcal{U}$ in $\mathsf{Syn}(\mathbb{T})$ is "logically generic" in the sense that it satisfies all and only the $\mathbb{T}$-provable equations, as in* (1.7).

Proof. For the proof, one shows that every term t is interpreted in $\mathcal{U}$ by "itself", *i.e.* by its own equivalence class under $\mathbb{T}$-provable equality,

$$(\, x_1, \ldots, x_m \mid t\,)^{\mathcal{U}} = [\, x_1, \ldots, x_m \mid t\,]$$

This is a simple induction on the construction of t, where the base case is given by (1.6). □

Even more important than being logically generic, though, is the following *universal property* of the syntactic model $\mathcal{U}$ in $\mathsf{Syn}(\mathbb{T})$.

> *Any model $\mathcal{M}$ in any finite product category $\mathcal{C}$ is the image of $\mathcal{U}$ under an essentially unique, finite product preserving functor $\mathcal{M}^\sharp : \mathsf{Syn}(\mathbb{T}) \to \mathcal{C}$,*
>
> $$\mathcal{M}^\sharp(\mathcal{U}) \cong \mathcal{M}\,.$$

(See Definition 1.1.20 below for a more precise formulation.) In this sense, the syntactic category $\mathsf{Syn}(\mathbb{T})$ may be thought of as the "free finite product category with a model of $\mathbb{T}$". To show this formally, first observe that any FP-functor $F : \mathsf{Syn}(\mathbb{T}) \to \mathcal{C}$ takes the syntactic model $\mathcal{U}$ in $\mathsf{Syn}(\mathbb{T})$ to a model $F\mathcal{U}$ in $\mathcal{C}$, with underlying interpretations

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

Indeed, that is true for any FP-category $\mathcal{S}$ in place of $\mathsf{Syn}(\mathbb{T})$ and any model in $\mathcal{S}$. Similarly, any natural transformation $\vartheta : F \to G$ between FP-functors determines a homomorphism of models $h = \vartheta_{\mathcal{U}} : F\mathcal{U} \to G\mathcal{U}$. In more detail, suppose $f : U \times U \to U$ is a basic operation, then there is a commutative diagram,

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

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

$$\mathsf{eval}_{\mathcal{U}} : \mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}), \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.8}$$

from the category of finite product preserving functors $\mathsf{Syn}(\mathbb{T}) \to \mathcal{C}$, with natural transformations as arrows, into the category of $\mathbb{T}$-models in $\mathcal{C}$. Indeed, this much is also true for any model in any FP-category $\mathcal{S}$; what is special about $\mathcal{U}$ is the following.

Proposition 1.1.18. *The functor* (1.8) *is an equivalence of categories, natural in $\mathcal{C}$.*

Proof. Let $\mathcal{M}$ be any model in an FP-category $\mathcal{C}$. Then the underlying interpretation of $\mathcal{M}$ is an assignment $f \mapsto f^{\mathcal{M}}$ for $f \in \Sigma$, which determines a functor $\mathcal{M}^\sharp : \mathsf{Syn}(\mathbb{T}) \to \mathcal{C}$, defined on objects by

$$\mathcal{M}^\sharp[x_1, \ldots, x_k] = |\mathcal{M}|^k$$

and on morphisms by

$$\mathcal{M}^\sharp[\,t_1, \ldots, t_n\,] = \big({t_1}^{\mathcal{M}}, \ldots, {t_n}^{\mathcal{M}}\big)\,.$$

In more detail, $\mathcal{M}^\sharp$ is defined on a morphism

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

in $\mathsf{Syn}(\mathbb{T})$ by the following rules:

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

2. The morphism
$$[x_1, \ldots, x_k \mid f(t_1, \ldots, t_m)] : [x_1, \ldots, x_k] \to [x_1]$$
is mapped to the composite
$$M^k \xrightarrow{\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_m\big)} M^m \xrightarrow{f^{\mathcal{M}}} M$$
where the $\mathcal{M}^\sharp t_i : M^k \to M$ are the values of $\mathcal{M}^\sharp$ on the morphisms $[t_i] : [x_1, \ldots, x_k] \to [x_i]$, for $i = 1, \ldots, m$, and $f^{\mathcal{M}}$ 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 $\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_n\big)$ where the $\mathcal{M}^\sharp t_i$ are the values of $\mathcal{M}^\sharp$ on the morphisms $[t_i] : [x_1, \ldots, x_k] \to [x_i]$, and
$$\big(\mathcal{M}^\sharp t_1, \ldots, \mathcal{M}^\sharp t_n\big) : M^k \longrightarrow M^n$$
is the evident n-tuple in the FP-category $\mathcal{C}$.

That $\mathcal{M}^\sharp : \mathsf{Syn}(\mathbb{T}) \to \mathcal{C}$ really is a functor 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 these specifications are well-defined on equivalence classes. Here we use the *soundness* of equational deduction with respect to models in FP categories.

Note that the functor $\mathcal{M}^\sharp$ is defined in such a way that it obviously preserves finite products, and that there is an isomorphism of models,

$$\mathcal{M}^\sharp(\mathcal{U}) \cong \mathcal{M}.$$

Thus we have shown that the functor "evaluation at $\mathcal{U}$",

$$\mathsf{eval}_{\mathcal{U}} : \mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}), \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{1.9}$$

is essentially surjective on objects, since $\mathsf{eval}_{\mathcal{U}}(\mathcal{M}^{\sharp}) = \mathcal{M}^{\sharp}(\mathcal{U}) \cong \mathcal{M}$.

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

Exercise 1.1.19. Verify this. (Hint: A homomorphism is entirely determined by what it does to the underlying object, and a natural transformation between FP functors is similarly determined by its component at $[x_1]$.)

Finally, naturality in $\mathcal{C}$ means the following. Suppose $\mathcal{M}$ 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 $\mathcal{M}$ to a model $F(\mathcal{M})$ in $\mathcal{D}$. Just as for the special case of $\mathcal{U}$, the interpretation is given by setting $f^{F(\mathcal{M})} = F(f^{\mathcal{M}})$ 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 (1.8), we mean that the following square commutes up to natural isomorphism:

$$\begin{tikzcd}
\mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}), \mathcal{C}) \arrow[r, "\mathsf{eval}_{\mathcal{U}}"] \arrow[d, "\mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}), F)"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \arrow[d, "\mathsf{Mod}(\mathbb{T}, F)"] \\
\mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}), \mathcal{D}) \arrow[r, "\mathsf{eval}_{\mathcal{U}}"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{C})
\end{tikzcd} \tag{1.10}$$

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

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

□

The equivalence of categories

$$\mathsf{Hom}_{\mathrm{FP}}(\mathsf{Syn}(\mathbb{T}),\mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T},\mathcal{C}) \tag{1.11}$$

actually determines $\mathsf{Syn}(\mathbb{T})$ and the universal model $\mathcal{U}$ uniquely, up to equivalence of categories and isomorphism of models. Indeed, to recover $\mathcal{U}$, just put $\mathsf{Syn}(\mathbb{T})$ for $\mathcal{C}$ and the identity functor $1_{\mathsf{Syn}(\mathbb{T})}$ on the left, to get $\mathcal{U}$ in $\mathsf{Mod}(\mathbb{T},\mathsf{Syn}(\mathbb{T}))$ on the right! To see that $\mathsf{Syn}(\mathbb{T})$ itself is also determined, observe that (1.11) says that the functor $\mathsf{Mod}(\mathbb{T},\mathcal{C})$ is *representable*, with representing object $\mathsf{Syn}(\mathbb{T})$, in an appropriate (i.e. bicategorical) sense. As usual, this fact can also be formulated in elementary terms as a universal mapping property of $\mathsf{Syn}(\mathbb{T})$, as follows:

Definition 1.1.20. The *classifying category* of an algebraic theory $\mathbb{T}$ is an FP-category $\mathcal{C}_{\mathbb{T}}$ with a distinguished model $\mathcal{U}$, called the *universal model*, such that:

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

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

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

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

$$\vartheta_{\mathcal{U}} = h.$$

Observe that (i) says that the evaluation functor (1.8) 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 $(\mathcal{C},\mathcal{U})$ and $(\mathcal{D},\mathcal{V})$ are both classifying categories for the same theory, then there are classifying functors,

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

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

We have now shown not only that every algebraic theory has a classifying category $\mathcal{C}_{\mathbb{T}}$, but also that the syntactic category $\mathsf{Syn}(\mathbb{T})$ is such a classifying category, and that it is essentially determined by that property. We record this as the following.

Theorem 1.1.21. *Every algebraic theory $\mathbb{T}$ has a classifying category $\mathcal{C}_{\mathbb{T}}$, which can be constructed as the syntactic category $\mathsf{Syn}(\mathbb{T})$ of $\mathbb{T}$, in the sense of Definition 1.1.15.*

Example 1.1.22. 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 $\mathbb{G} = \mathsf{syn}(\mathbb{T}_{\mathsf{Group}})$ for the syntactic category, which has contexts $[x_1, \ldots, x_n]$ as objects, and terms built from variables and the group operations (modulo renaming of variables and provability from the group laws) as arrows. A finite product preserving functor $G : \mathbb{G} \to \mathsf{Set}$ is determined uniquely, up to natural isomorphism, by its action on the context $[x_1]$ and the terms representing the basic operations. If we set

$$\begin{aligned} |\mathcal{G}| &:= G[x_1] \, , & u^{\mathcal{G}} &:= G(\cdot \mid e) \, , \\ i^{\mathcal{G}} &:= G(x_1 \mid {x_1}^{-1}) \, , & m_G &= G(x_1, x_2 \mid x_1 \cdot x_2) \, , \end{aligned}$$

then $\mathcal{G} = (|\mathcal{G}|, u^{\mathcal{G}}, i^{\mathcal{G}}, m^{\mathcal{G}})$ is just a group, with unit $u^{\mathcal{G}}$, inverse $i^{\mathcal{G}}$, and multiplication $m^{\mathcal{G}}$. That the interpretation $\mathcal{G}$ satisfies the group equaitons follows from the fact that $\mathcal{G}$ does (it is generic by Proposition 1.1.17), the preservation of finite products by G, and its functoriality, which implies preservation of the corresponding commutative diagrams.

Conversely, any group $\mathcal{G} = (G, u, i, m)$ determines a finite product preserving functor $\mathcal{G}^\sharp : \mathbb{G} \to \mathsf{Set}$, by setting $\mathcal{G}^\sharp[x_1] = G$, etc. Thus $\mathsf{Mod}(\mathbb{G}, \mathsf{Set})$ 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 1.1.23. 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 (since products in functor categories are computed pointwise). The correspondence extends to natural transformations, giving the previously observed (Example 1.1.12) equivalance of categories,

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

1.1.4 Soundness and 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 $\mathcal{M}$ of the theory in any FP-category satisfies the equation, $\mathcal{M} \models s = t$. This is called the *soundness* of the equational calculus with respect to categorical models, and it can be shown by a straightforward induction on the equational proof that establishes $\mathbb{T} \vdash s = t$. The converse statement reads:

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

This is called *completeness*, and (together with soundness) it says that the equational calculus suffices for proving all (and only) the equations that hold generally in the semantics. For functorial semantics, this condition holds in an especially strong way: by Proposition

1.1.17, we already know that the syntactic model $\mathcal{U}$ in $\mathsf{Syn}(\mathbb{T})$ is logically generic, in the sense that satisfaction by $\mathcal{U}$ is equivalent to provability in $\mathbb{T}$,

$$\mathcal{U} \models s = t \qquad \Longleftrightarrow \qquad \mathbb{T} \vdash s = t.$$

But since $\mathsf{Syn}(\mathbb{T})$ is a classifying category for $\mathbb{T}$ and $\mathcal{U}$ is universal in the sense of Definition 1.1.20 it follows that we also have completeness:

Theorem 1.1.24 (Soundness and completeness of equational logic)**.** *For any terms s, t we have $\mathbb{T} \vdash s = t$ if and only if every model $\mathcal{M}$ in every FP-category $\mathcal{C}$ satisfies $s = t$.*

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

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

and so from $s^{\mathcal{U}} = t^{\mathcal{U}}$ we get $s^{\mathcal{M}} = t^{\mathcal{M}}$.

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

Classically, it is seldom the case that there exists a generic model; instead, one usually considers completeness with respect to a class of special models, say, those in Set. Completeness with respect to a restricted class of models is of course a stronger statement than completeness with respect to all models in all categories; indeed, one need only test an equation in the restricted class to know that it can be proved, and therefore holds in all models. Toward the classical result, we can first consider completeness with respect to just "variable models" in Set, i.e. in arbitrary functor categories $\mathsf{Set}^{\mathbb{C}}$. That result follows immediately from the next lemma.

Lemma 1.1.25. *Let $\mathbb{T}$ be an algebraic theory. The Yoneda embedding*

$$\mathsf{y} : \mathcal{C}_{\mathbb{T}} \to \widehat{\mathcal{C}_{\mathbb{T}}} = \mathsf{Set}^{\mathcal{C}_{\mathbb{T}}^{\mathsf{op}}}$$

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

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

Example 1.1.26. We consider group theory one more time. We again write simply $\mathbb{G}$ for the syntactic (classifying) category of the theory $\mathbb{T}_{\mathsf{Group}}$ of groups. As a presheaf on $\mathbb{G}$, the generic group $\mathcal{Y} \in \widehat{\mathbb{G}}$ satisfies every equation that is satisfied by all groups, and no others.

Let us describe its underlying object $Y = |\mathcal{Y}|$ explicitly as a “variable set”. By definition, the presheaf Y is represented by the underlying object $U = |\mathcal{U}|$ of the universal group in $\mathbb{G}$, which in syntactic terms is the context with one variable,

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

The values of this functor thus comprise a family of sets parametrized by the objects $[x_1, \ldots, x_n]$ of $\mathbb{G}$; namely, for every $n \in \mathbb{N}$, we have the set

$$Y_n = \mathbb{G}([x_1, \ldots, x_n], [x_1])$$

consisting of all (equivalence classes of) terms $[\, x_1, \ldots, x_n \mid t \,]$ 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

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

Moreover, the action of the functor Y on a map

$$s : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n] \qquad \text{in } \mathbb{G}$$

can be described by substitution of the terms $s = (s_1, \ldots, s_n)$ into the elements $t \in Y_n$,

$$Y(s)(t) = \mathbb{G}(s, [x_1])(t) = t[\, s_1/x_1, \ldots, s_n/x_n \,] \, .$$

In terms of the free groups $F(n)$, the terms $s_1, \ldots, s_n$ in the context $x_1, \ldots, x_m$ are elements of the free group $F(m)$, and so they determine a unique homomorphism

$$\overline{s} : F(n) \cong F(1) + ... + F(1) \longrightarrow F(m)$$

such that $\overline{s}(x_i) = s_i$ for $i = 1, ..., n$. Composition with $\overline{s} : F(n) \to F(m)$ then encodes the corresponding substitution, in the sense that the following diagram commutes (as the reader should verify!).

$$\begin{tikzcd}
{[\, x_1, \ldots, x_n \,]} & \mathbb{G}([\, x_1, \ldots, x_n \,], [x_1]) \arrow[r, "\cong"] \arrow[d, "\mathbb{G}(s, [x_1])"'] & \mathsf{Group}(F(1), F(n)) \arrow[d, "\mathsf{Group}(F(1), \overline{s})"] & F(n) \arrow[d, "\overline{s}"] \\
{[\, x_1, \ldots, x_m \,]} \arrow[u, "s"] & \mathbb{G}([\, x_1, \ldots, x_m \,], [x_1]) \arrow[r, "\cong"'] & \mathsf{Group}(F(1), F(m)) & F(m)
\end{tikzcd} \tag{1.12}$$

Finally, the unit, inverse, and multiplication operations of the internal group $\mathcal{Y}$ are determined at each stage Y_n by the corresponding operations on the free group $F(n)$ (as the reader should verify!). We will discover a deeper reason for this in Section 1.2.1.

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

Lemma 1.1.27. *For any small category $\mathbb{C}$, there is a jointly faithful family $(E_i)_{i\in I}$ of FP-functors $E_i : \mathsf{Set}^{\mathbb{C}} \to \mathsf{Set}$, with I a set. That is, for any maps $f, g : A \to B$ in $\mathsf{Set}^{\mathbb{C}}$, if $E_i(f) = E_i(g)$ for all $i \in I$, then $f = g$.*

Proof. We can take $I = \mathbb{C}_0$, the set of objects of $\mathbb{C}$, and the evaluation functors

$$E_c = \mathsf{eval}_c : \mathsf{Set}^{\mathbb{C}} \to \mathsf{Set}\,,$$

for all $c \in \mathbb{C}$. These are clearly jointly faithful. Note that they also preserve all limits and colimits, since these are constructed pointwise in functor categories. □

Proposition 1.1.28. *Suppose $\mathbb{T}$ is an algebraic theory. For any terms s, t,*

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

Thus the equational logic of algebraic theories is sound and complete with respect to Set-valued semantics.

Proof. Combine the foregoing lemma with the fact, from Lemma 1.1.25, that the Yondea embedding is a generic model. □

The completeness of equational reasoning was originally proved by Birkhoff [Bir35]. The proof is not particularly difficult; we have chosen to redo it in this way because the method will generalize to other systems of logic in later chapters.

Exercise 1.1.29. We described the functor $Y = \mathsf{y}U : \mathbb{G}^{\mathsf{op}} \to \mathsf{Set}$ represented by the underlying object $U = [x_1]$ of the universal group $\mathcal{U}$ in terms of the free groups $F(n)$. Verify that the action of Y on the arrows of $\mathbb{G}$ is indeed given by substitution of terms by checking that diagram (1.12) commutes. Also describe the group structure on Y in $\widehat{\mathbb{G}}$ explicitly in terms of that on the free groups.

Exercise 1.1.30. Let $t = t(x_1, \dots, x_n)$ be a term of group theory in the variables $x_1, \dots, x_n$. On the one hand we can think of t as an element of the free group $F(n)$, and on the other we can consider the interpretation of t with respect to the representable group $\mathcal{Y}$ in $\widehat{\mathbb{G}}$, namely as a natural transformation $t^{\mathcal{Y}} : Y^n \to Y$. Suppose $s = s(x_1, \dots, x_n)$ is another such term in the same variables $x_1, \dots, x_n$. Show that $s^{\mathcal{Y}} = t^{\mathcal{Y}}$ 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 so far. We have reformulated certain traditional *logical* notions in terms of *categorical* ones. The traditional approach may be described as involving the following four different parts:

Terms

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 a *signature* consisting of some basic operation symbols.

Equations

Algebraic theories have a particularly simple *logic* that involves only equations between terms and equational reasoning, which is basically *substitution* of equals for equals and the laws of an *equivalence relation*.

Theories

An *algebraic theory* then consists of a signature and a set of *axioms*, which are just equations between terms. Such theories are regarded as *logical syntax*: sets of uninterpreted, formal expressions, generated inductively by rules of inference.

Models

An algebraic theory can be modeled by a *set* equipped with some *operations* interpreting the signature. Such an interpretation is a *model* if it satisfies the axioms of the theory, meaning that the functions interpreting the terms that occur in the equational axioms are actually equal.

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

Theories are categories

From a given theory we construct a structured category that captures the same information in a way that is independent of a particular presentation by basic operations and axioms.

Models are functors

A model is a structure-preserving functor from the theory to a category with the same structure. For algebraic theories, a model is a functor that preserves finite products, which 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 homomorphism of models for free: since models are functors, the homomorphisms between them are just the natural transformations. Such homomorphisms agree with the usual notion, consisting of a function on the underlying sets that "respects" the algebraic structure.

Universal models

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

Logical completeness

The construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ from the syntax of the theory $\mathbb{T}$ shows that the universal model is also *generic*: it has exactly those properties that are provable in the theory $\mathbb{T}$. This implies the *soundness and completeness* of the logic with respect to general categorical semantics. Completeness with respect to a restricted class of models, such as those in Set, then results from an embedding theorem for the classifying category.

1.2 Lawvere duality

The scheme of functorial semantics outlined in the previous section applies to many other systems of logic than algebraic theories, some of which will be considered in later chapters. A further aspect of this approach is especially transparent in the case of algebraic theories; namely, a deep and fascinating *duality* relating syntax and semantics. We devote the rest of this chapter to its investigation.

1.2.1 Logical duality

There is a remarkable and far-reaching duality in logic of the form

$$\text{Syntax} \ \simeq \ \text{Semantics}^{\mathsf{op}}.$$

It was discovered by F.W. Lawvere in the 1960s and presented in some early papers, [Law63a, Law63b, Law65, Law69], but it has still hardly been noticed in conventional logic—perhaps because its recognition requires the tools of category theory.

We can see this duality quite clearly in the case of algebraic theories. Let $\mathcal{C}_{\mathbb{T}}$ be the classifying category for an equational theory $\mathbb{T}$, like the theory of groups, constructed syntactically as in section 1.1.2 above. So the objects of $\mathcal{C}_{\mathbb{T}}$ are contexts of variables $[x_1 \dots, x_n]$, up to renaming, and the arrows $(t_1, ..., t_n) : [x_1 \dots, x_m] \to [x_1 \dots, x_n]$ are n-tuples of terms in context $[x_1 \dots, x_m \mid t_i]$, up to $\mathbb{T}$-provable equality. We will see that this syntactic category $\mathcal{C}_{\mathbb{T}}$ is in fact dual to a certain subcategory of *models* of $\mathbb{T}$ (in Set). Specifically, there is a small, full subcategory $\mathsf{mod}(\mathbb{T}) \hookrightarrow \mathsf{Mod}(\mathbb{T})$ and an equivalence of categories,

$$\mathcal{C}_{\mathbb{T}} \simeq \mathsf{mod}(\mathbb{T})^{\mathsf{op}},$$

making the *syntactic* category $\mathcal{C}_{\mathbb{T}}$ dual to a subcategory of the *semantic* category $\mathsf{Mod}(\mathbb{T})$. Thus, in particular, there is an invariant representation of the syntax of the theory $\mathbb{T}$ "hidden" inside the category of models of $\mathbb{T}$.

Indeed, it is quite easy to specify $\mathsf{mod}(\mathbb{T})$ explicitly: it is the full subcategory of $\mathsf{Mod}(\mathbb{T})$ on the *finitely generated free models* $F(n)$,

$$\mathsf{mod}(\mathbb{T})_0 = \left\{ F(n) \mid n \in \mathbb{N} \right\}.$$

We will have a more intrinsic characterization by the end of this chapter.

Theorem 1.2.1. *Let $\mathbb{T}$ be an algebraic theory, and let*

$$\mathsf{mod}(\mathbb{T}) \hookrightarrow \mathsf{Mod}(\mathbb{T})$$

be the full subcategory of finitely generated free models of $\mathbb{T}$. Then $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ classifies $\mathbb{T}$ models. That is to say, for any FP-category $\mathcal{C}$, there is an equivalence of categories,

$$\mathsf{Hom}_{\mathrm{FP}}(\mathsf{mod}(\mathbb{T})^{\mathsf{op}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C}), \tag{1.13}$$

which is natural in $\mathcal{C}$.

Before giving the (somewhat lengthy, but straightforward) proof of the theorem, let us observe that the syntax-semantics duality follows immediately. Indeed, given (1.13), there is then an equivalence,

$$\mathcal{C}_{\mathbb{T}} \simeq \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \tag{1.14}$$

between the (syntactically constructed) classifying category $\mathcal{C}_{\mathbb{T}}$ and the opposite of the (semantic) category $\mathsf{mod}(\mathbb{T})$ of finitely generated free models, because by Proposition 1.1.18, both categories $\mathcal{C}_{\mathbb{T}}$ and $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ represent the same functor $\mathsf{Mod}(\mathbb{T}, \mathcal{C})$.

Proof of Theorem 1.2.1. First, observe that $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ has all finite products, since $\mathsf{mod}(\mathbb{T})$ has all finite coproducts. Indeed, for the finitely generated free algebras $F(n)$ we have

$$\begin{aligned} F(n) + F(m) &\cong F(n+m), \\ 0 &\cong F(0), \end{aligned}$$

in $\mathsf{Mod}(\mathbb{T})$, since the left adjoint F preserves all colimits.

For the universal $\mathbb{T}$-algebra $\mathcal{U}$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, let

$$U = F(1),$$

so that every object in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ is indeed a power of U,

$$F(n) \cong U^n.$$

We next interpret the signature $\Sigma_{\mathbb{T}}$. For each basic operation symbol $f \in \Sigma_{\mathbb{T}}$, with arity k, there is an element of the free algebra $F(k)$ built from the operation $f^{F(k)} : F(k)^k \to F(k)$ and the k generators $x_1, \ldots, x_k \in F(k)$, namely

$$f^{F(k)}(x_1, \ldots, x_k).$$

E.g. in the theory of groups, there is the element $x \cdot y$ in the free group on the two generators x, y. By freeness of $F(1)$, each element $t \in F(k)$ determines a unique homomorphism $\overline{t} : F(1) \to F(k)$ in $\mathsf{mod}(\mathbb{T})$ taking the generator $x \in F(1)$ to $t = \overline{t}(x)$. Thus associated to the element $f^{F(k)}(x_1, \ldots, x_k) \in F(k)$ there is a homomorphism

$$\overline{f^{F(k)}(x_1, \ldots, x_k)} : F(1) \longrightarrow F(k) \qquad \text{in } \mathsf{mod}(\mathbb{T}).$$

We take this map, regarded as an arrow in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, as the $\mathcal{U}$-interpretation of the basic operation symbol f,

$$f^{\mathcal{U}} := \overline{f^{F(k)}(x_1, \dots, x_k)} : U^k \longrightarrow U \qquad \text{in } \mathsf{mod}(\mathbb{T})^{\mathsf{op}}.$$

It then follows easily that for any term in context $(x_1 \dots, x_k \mid t)$, the interpretation

$$[x_1 \dots, x_k \mid t]^{\mathcal{U}} : U^k \longrightarrow U$$

will be the unique homomorphism $\overline{t^{F(k)}} : F(1) \to F(k)$ corresponding to the element $t^{F(k)} \in F(k)$ (proof by induction!).

Moreover, for every axiom $(s = t)$ of $\mathbb{T}$, we then have $\mathcal{U} \models s = t$. Indeed,

$$[x_1 \dots, x_k \mid s]^{\mathcal{U}} = [x_1 \dots, x_k \mid t]^{\mathcal{U}} : U^k \longrightarrow U$$

if, and only if, the corresponding homomorphisms $\overline{s}, \overline{t} : F(1) \to F(k)$ agree, which they do just if the associated elements of the free algebra $F(k)$ agree, by the freeness of $F(1)$. And the latter holds, in turn, simply because $F(k)$ is a $\mathbb{T}$-algebra. Indeed, consider the example of the two generators x, y of the free *abelian* group $F(2)$, for which we have $x \cdot y = y \cdot x$ simply because $F(2)$ is abelian. Thus we indeed have a $\mathbb{T}$-model $\mathcal{U}$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, consisting of the free algebras.

We next show that $\mathcal{U}$ has the required universal property, in three steps:

Step 1. Let $\mathcal{A}$ be any $\mathbb{T}$-algebra in Set. Then there is a product-preserving functor,

$$\mathcal{A}^\sharp : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}$$

with $\mathcal{A}^\sharp(\mathcal{U}) \cong \mathcal{A}$ (as $\mathbb{T}$-models), namely:

$$\mathcal{A}^\sharp(-) = \mathsf{Hom}_{\mathsf{Mod}(\mathbb{T})}(-, \mathcal{A}),$$

where we of course restrict the representable functor $\mathsf{Hom}_{\mathsf{Mod}(\mathbb{T})}(-, \mathcal{A}) : \mathsf{Mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}$ along the (full) inclusion

$$\mathsf{mod}(\mathbb{T}) \hookrightarrow \mathsf{Mod}(\mathbb{T})$$

of the finitely generated, free algebras. The (restricted) functor

$$\mathcal{A}^\sharp : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}$$

clearly preserves products: for each object $U^n \in \mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, we have

$$\mathcal{A}^\sharp(U^n) = \mathsf{Hom}_{\mathsf{Mod}(\mathbb{T})}(F(n), \mathcal{A}) \cong \mathsf{Hom}_{\mathsf{Set}}(\, n, |\mathcal{A}|\,) \cong A^n$$

and, in particular, $\mathcal{A}^\sharp(U) \cong A$.

Finally, let us show that for any basic operation symbol f, we have $\mathcal{A}^\sharp(f^\mathcal{U}) = f^\mathcal{A}$, up to isomorphism. Indeed, given any algebra $\mathcal{A}$ and operation $f^\mathcal{A} : A^n \to A$, we have a commutative diagram,

$$\begin{tikzcd}
A^n \arrow[rr, "f^{\mathcal{A}}"] \arrow[d, "\cong"'] & & A \arrow[d, "\cong"] \\
\mathsf{Hom}(F(n), \mathcal{A}) \arrow[rr, "f^*"'] & & \mathsf{Hom}(F(1), \mathcal{A})
\end{tikzcd} \tag{1.15}$$

where f^* is precomposition with the homomorphism

$$\begin{tikzcd}
F(n) & & F(1) \arrow[ll, "\overline{f^{F(n)}(x_1, \dots, x_n)}"']
\end{tikzcd}$$

To see that (1.15) commutes, take any $(a_1, \dots, a_n) \in A^n$ with associated homomorphism $\overline{(a_1, \dots, a_n)} : F(n) \to \mathcal{A}$ and precompose with $\overline{f^{F(n)}(x_1, \dots, x_n)}$ to get a map $F(1) \to \mathcal{A}$, picking out the element

$$\begin{aligned}
\overline{(a_1, \dots, a_n)} \circ \overline{f^{F(n)}(x_1, \dots, x_n)}(x) &= \overline{(a_1, \dots, a_n)}(f^{F(n)}(x_1, \dots, x_n)) \\
&= \overline{(a_1, \dots, a_n)} \circ f^{F(n)}(x_1, \dots, x_n) \\
&= f^\mathcal{A} \circ \overline{(a_1, \dots, a_n)}(x_1, \dots, x_n) \\
&= f^\mathcal{A}(a_1, \dots, a_n)
\end{aligned}$$

where x is the generator of $F(1)$, and we have used the fact that $\overline{(a_1, \dots, a_n)}$ is a homomorphism and therefore commutes with the respective interpretations of f.

But now note that

$$\begin{tikzcd}
F(n) & & F(1) \arrow[ll, "\overline{f^{F(n)}(x_1, \dots, x_n)}"']
\end{tikzcd}$$

in $\mathsf{mod}(\mathbb{T})$ is

$$\begin{tikzcd}
U^n \arrow[r, "f^U"'] & U
\end{tikzcd}$$

in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, and that $\mathsf{Hom}(F(n), \mathcal{A}) = \mathcal{A}^\sharp(U^n)$ and $f^* = \mathcal{A}^\sharp(f^U)$. Thus (1.15) shows that indeed $\mathcal{A}^\sharp(f^\mathcal{U}) = f^\mathcal{A}$, up to isomorphism. Thus we indeed have $\mathcal{A}^\sharp(\mathcal{U}) \cong \mathcal{A}$ as algebras, as required.

We leave it to the reader to verify that any homomorphism $h : F(\mathcal{U}) \to G(\mathcal{U})$ of $\mathbb{T}$-algebras $F(\mathcal{U}), G(\mathcal{U})$ arising from FP-functors $F, G : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}$ is of the form $h = \vartheta_\mathcal{U}$ for a unique natural transformation $\vartheta : F \to G$.

Exercise 1.2.2. Show this.

Step 2. Let $\mathbb{C}$ be any (locally small) category, and $\mathcal{A}$ a $\mathbb{T}$-algebra in $\mathsf{Set}^{\mathbb{C}}$. Using the isomorphism

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

each $\mathcal{A}(C)$ is a $\mathbb{T}$-algebra (in Set), which by Step 1 has a classifying functor,

$$\mathcal{A}(C)^{\sharp} : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}.$$

Together, these determine a single functor $\mathcal{A}^{\sharp} : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}^{\mathbb{C}}$, defined on any $U^n \in \mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ by

$$(\mathcal{A}^{\sharp}(U^n))(C) \cong \mathcal{A}(C)^{\sharp}(U^n) \cong (AC)^n.$$

The action on arrows $U^n \to U^m$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ is similarly determined pointwise by the components

$$(\mathcal{A}^{\sharp}(U^n))(C) \cong \mathcal{A}(C)^{\sharp}(U^n) \to \mathcal{A}(C)^{\sharp}(U^m) = (\mathcal{A}^{\sharp}(U^m))(C),$$

for all $C \in \mathbb{C}$.

In this way, we have an FP-functor $\mathcal{A}^{\sharp} : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}^{\mathbb{C}}$, and an isomorphism of models $\mathcal{A}^{\sharp}(\mathcal{U}) \cong \mathcal{A}$ in $\mathsf{Set}^{\mathbb{C}}$. It is then clear that any natural transformation $\mathcal{A}^{\sharp} \to \mathcal{B}^{\sharp}$ gives rise to a homomorphism $\mathcal{A}^{\sharp}(\mathcal{U}) \to \mathcal{B}^{\sharp}(\mathcal{U})$, and that the resulting functor

$$\mathsf{Hom}_{\mathrm{FP}}(\mathsf{mod}(\mathbb{T})^{\mathsf{op}}, \mathsf{Set}^{\mathbb{C}}) \to \mathsf{Mod}(\mathbb{T}, \mathsf{Set}^{\mathbb{C}})$$

is an equivalence.

Step 3. For the general case, let $\mathcal{C}$ be any (locally small) FP-category, and A a $\mathbb{T}$-algebra in $\mathcal{C}$. Use the Yoneda embedding

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

to send A to an algebra $\mathcal{A} = \mathsf{y}(\mathsf{A})$ in $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$ (since y preserves finite products). Now apply Step 2 to get a classifying functor,

$$\mathcal{A}^{\sharp} : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}.$$

We claim that $\mathcal{A}^{\sharp}$ factors through the Yoneda embedding by an FP-functor $\mathsf{A}^{\sharp}$,

$$\begin{tikzcd}
 & \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}} \\
\mathsf{mod}(\mathbb{T})^{\mathsf{op}} \arrow[ur, "\mathcal{A}^{\sharp}"] \arrow[r, dotted, "\mathsf{A}^{\sharp}"'] & \mathcal{C}. \arrow[u, hook, "\mathsf{y}"']
\end{tikzcd}$$

Indeed, we know that the objects of $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ all have the form U^n, and for their images we have

$$\mathcal{A}^{\sharp}(U^n) \cong \mathcal{A}^{\sharp}(U)^n \cong \mathsf{y}(\mathsf{A})^n \cong \mathsf{y}(\mathsf{A}^n)\,.$$

Thus the images of the objects of $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$ are all representable. Since y is full and faithful, the claim is established, and the resulting functor $\mathsf{A}^\sharp : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathcal{C}$ preserves finite products because $\mathcal{A}^\sharp$ does so, and y creates them. Clearly,

$$\mathsf{A}^\sharp(\mathcal{U}) \cong \mathsf{A},$$

since y reflects isos.

Naturality of the equivalence

$$\mathsf{Hom}_{\mathrm{FP}}(\mathsf{mod}(\mathbb{T})^{\mathsf{op}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C}),$$

in $\mathcal{C}$ is essentially automatic, using the fact that it is induced by evaluating an FP functor $F : \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \to \mathcal{C}$ at the universal model $\mathcal{U}$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$. □

As already mentioned, since the classifying category is uniquely determined, up to equivalence, by its universal property, combining the foregoing theorem with the syntactic construction of $\mathcal{C}_{\mathbb{T}}$ given in theorem 1.1.18 yields the following:

Corollary 1.2.3 (Logical duality for algebraic theories)**.** *For any algebraic theory $\mathbb{T}$, there is an equivalence of categories,*

$$\mathsf{Syn}(\mathbb{T}) \;\simeq\; \mathcal{C}_{\mathbb{T}} \;\simeq \mathsf{mod}(\mathbb{T})^{\mathsf{op}} \tag{1.16}$$

between the classifying category $\mathcal{C}_{\mathbb{T}}$ constructed syntactically as $\mathsf{Syn}(\mathbb{T})$ and the semantic construction as the opposite of the category $\mathsf{mod}(\mathbb{T})$ of finitely generated, free models.

Thus, as claimed, the construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ from the syntax of $\mathbb{T}$, on the one hand, and its semantic construction as $\mathsf{mod}(\mathbb{T})$, taken together, imply that there is an invariant representation of the *syntax* of $\mathbb{T}$ hidden, as it were, in the opposite of the *semantics*, namely the category $\mathsf{Mod}(\mathbb{T})$ of all $\mathbb{T}$-models. The reader may wish to reflect on the importance of (i) considering the *category* of all models, rather than the mere collection of them, and (ii) generalizing from set-theoretic to categorical models, in arriving at the fundamental logical duality expressed by (1.16).

In section 1.2.5 below, we shall consider how to actually *recover* this syntactic category $\mathcal{C}_{\mathbb{T}}$ from the semantic category $\mathsf{Mod}(\mathbb{T})$ by identifying the subcategory $\mathsf{mod}(\mathbb{T})$ intrinsically; indeed, it will be seen to consist of certain continuous functors $\mathsf{Mod}(\mathbb{T}) \to \mathsf{Set}$. Before doing this, however, let us examine the fundamental equivalence (1.16) explicitly in a few special cases.

Example 1.2.4. Consider the trivial theory $\mathbb{T}_0$ of Example 1.1.4, with no basic operations or equations. A model of $\mathbb{T}_0$ in Set is just a set X, equipped with no operations, and satisfying no further conditions (and similarly in any other FP category). All $\mathbb{T}_0$-algebras are free, and the finitely generated ones are just the finite sets, thus

$$\mathsf{mod}(\mathbb{T}_0) = \mathsf{Set}_{\mathrm{fin}}$$

is the category of finite sets (to be more specific, let us take a skeleton, with one n-element set $[n]$ for each $n \in \mathbb{N}$). Theorem 1.2.1 tells us that, for any FP category $\mathcal{C}$, there is an equivalence

$$\mathsf{Hom}_{\mathrm{FP}}(\mathsf{Set}_{\mathrm{fin}}^{\mathsf{op}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}_0, \mathcal{C}) \simeq \mathcal{C}.$$

This simply says that $\mathsf{Set}_{\mathrm{fin}}^{\mathsf{op}}$ is the free FP category on one object. Equivalently, $\mathsf{Set}_{\mathrm{fin}}$ is the free finite *coproduct* category on one object. And this is indeed the case, as one can easily see directly (the objects are the finite cardinal numbers $[0], [1], [2] = [1] + [1], \dots$).

The logical duality of corollary 1.2.3 now tells us that the dual of the category of finite sets is the *syntactic category* of $\mathbb{T}_0$,

$$\mathcal{C}_{\mathbb{T}_0} \;\simeq\; \mathsf{Set}_{\mathrm{fin}}^{\mathsf{op}}.$$

Let us see how the syntax of the pure theory of equality $\mathbb{T}_0$ is "hidden" in the opposite of the category of finite sets. The only terms in context are the variables $(\, x_1, ..., x_n \mid x_i\,)$, representing the product projections, and the provable equations are just those that are true of them *as terms*, so $x_i = x_j$ just if $i = j$. The maps $[n] \to [k]$ in $\mathcal{C}_{\mathbb{T}_0}$ are therefore just tuples of variables,

$$(\, x_{i_1}, ..., x_{i_k}\,) : [\, x_1, \dots, x_n\,] \longrightarrow [\, x_1, \dots, x_k\,]$$

Our corollary tells us that this is the category of finite sets, which we can see immediately by reading the contexts $[x_1, \dots, x_n]$ as coproducts $1 + \cdots + 1$ and a tuple such as $(x_2, x_5) : [x_1, \dots, x_5] \to [x_1, x_2]$ as a *cotuple* like $[i_2, i_5] : 1_1 + 1_2 \to 1_1 + ... + 1_5$.

Example 1.2.5. For a less trivial example, consider the theory $\mathbb{T}_{\mathsf{Ab}}$ of abelian groups. Duality tells us that the syntactic category $\mathcal{C}_{\mathbb{T}_{\mathsf{Ab}}}$ is dual to the category of finitely generated, free abelian groups $\mathsf{Ab}_{\mathrm{fg}}$,

$$\mathcal{C}_{\mathbb{T}_{\mathsf{Ab}}} \simeq \mathsf{Ab}_{\mathrm{fgf}}^{\mathsf{op}}.$$

This gives us a representation of the syntax of (abelian) group theory in the category of abelian groups, which can be described concretely as follows, using fact that for Abelian groups A, B we have an isomorphism $A + B \cong A \times B$.

- The basic types of variables $[-] = 1,\ [x_1] = U,\ [x_1, x_2] = U \times U, \dots$ are represented by the free abelian groups $\{0\},\ \mathbb{Z},\ \mathbb{Z} + \mathbb{Z} = \mathbb{Z}^2,\ \mathbb{Z}^3, \dots$.

- The group unit $u : 1 \to U$ is the zero homomorphism $0 : \mathbb{Z} \to \{0\}$.

- The inverse operation $i : U \to U$ is the unique homomorphism $\mathbb{Z} \to \mathbb{Z}$ taking 1 to -1 (and therefore n to $-n$).

- The group operation $m : U \times U \to U$ is the homomorphism $\overline{+} : \mathbb{Z} \to \mathbb{Z} + \mathbb{Z}$ taking 1 to $\langle 1, 0\rangle + \langle 0, 1\rangle = \langle 1, 1\rangle$, (using $\mathbb{Z} + \mathbb{Z} \cong \mathbb{Z} \times \mathbb{Z}$).

- The laws of abelian groups (and no further ones!) hold under this interpretation, because by (1.15) the group structure on any abelian group A is induced by precomposing with these "co-operations". For instance, the multiplication $+^A : |A| \times |A| \to |A|$ works by first "classifying" a pair of elements $a, b \in |A|$ by the homomorphism $\overline{(a,b)} : \mathbb{Z} + \mathbb{Z} \to A$ and then precomposing with the comultiplication $\overline{+} : \mathbb{Z} \to \mathbb{Z} + \mathbb{Z}$ to obtain the homomorphism $\overline{a+b} : \mathbb{Z} \to A$, which classifies the element $a + b \in |A|$.

$$\begin{tikzcd}
\mathbb{Z}+\mathbb{Z} \arrow[r, "\overline{(a,b)}"] & A \\
\mathbb{Z} \arrow[u, "\overline{+}"] \arrow[ur, "\overline{a+b}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\hom(\mathbb{Z}+\mathbb{Z}, A) \arrow[r, "\cong"] \arrow[d, "\hom(\overline{+}, A)"'] & {|A| \times |A|} \arrow[d, "+^A"] \\
\hom(\mathbb{Z}, A) \arrow[r, "\cong"'] & {|A|}
\end{tikzcd}$$

Example 1.2.6. The category of *affine schemes* is, by definition, the dual of the category of commutative rings with unit,

$$\mathsf{Scheme}_{\mathrm{aff}} = \mathsf{Ring}^{\mathsf{op}}$$

There is therefore a ring object in affine schemes – called the *affine line* – based on the finitely generated free algebra $F(1) = \mathbb{Z}[x]$, the ring of polynomials in one variable x with integer coefficients. The "co-operations" of $+$ and $\cdot$ are given in rings by the homomorphisms $\mathbb{Z}[x] \to \mathbb{Z}[x, y]$ taking the generator x to the elements $x + y$ and $x \cdot y$.

Exercise 1.2.7. Prove directly that $\mathsf{Set}_{\mathrm{fin}}$ is the free finite coproduct category on one object.

Exercise 1.2.8. Show that for any algebraic theory $\mathbb{T}$, the forgetful functor $V : \mathsf{Mod}(\mathbb{T}) \to \mathsf{Set}$ underlies an algebra $\mathcal{V}$ in the functor category $\mathsf{Set}^{\mathsf{Mod}(\mathbb{T})}$. In more detail, each n-ary operation symbol f determines a natural transformation $f^{\mathcal{V}} : V^n \to V$, since the homomorphisms in $\mathsf{Mod}(\mathbb{T})$ commute with the respective operations interpreting f. Indeed, given any algebra $\mathcal{A}$ we have the underlying set $V(\mathcal{A}) = A$ and an operation $f^{\mathcal{A}} : A^n \to A$, and for every homomorphism $h : \mathcal{A} \to \mathcal{B}$ to another algebra $\mathcal{B}$, there is a commutative square,

$$\begin{tikzcd}
A^n \arrow[r, "h^n"] \arrow[d, "f^{\mathcal{A}}"'] & B^n \arrow[d, "f^{\mathcal{B}}"] \\
A \arrow[r, "h"'] & B.
\end{tikzcd} \tag{1.17}$$

So we can set $(f^{\mathcal{V}})_{\mathcal{A}} = f^{\mathcal{A}}$ to get a natural transformation $f^{\mathcal{V}} : V^n \to V$. Now check that this really is an algebra $\mathcal{V}$ in $\mathsf{Set}^{\mathsf{Mod}(\mathbb{T})}$.

Exercise 1.2.9. * Show that the algebra described in the previous exercise is represented by the universal one $\mathcal{U}$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}} \hookrightarrow \mathsf{Mod}(\mathbb{T})^{\mathsf{op}}$ under the (covariant) Yoneda embedding,

$$\mathsf{y} : \mathsf{Mod}(\mathbb{T})^{\mathsf{op}} \longrightarrow \mathsf{Set}^{\mathsf{Mod}(\mathbb{T})}.$$

1.2.2 Lawvere algebraic theories

Nothing in the foregoing account of the functorial semantics for algebraic theories really depended on the primarily syntactic nature of such theories, i.e. their specification in terms of operations and equations. We can thus generalize it to “abstract” algebraic theories, which can be regarded as a *presentation-free* notion of an algebraic theory.

Definition 1.2.10 (cf. Definition 1.1.2)**.** A *Lawvere algebraic theory* $\mathbb{A}$ is a small category with finite products, the objects of which form a sequence $A^0, A^1, A^2, \ldots$ with $A^0 = \mathsf{1}$ terminal and $A^{n+1} = A^n \times A^1$ for all $n \in \mathbb{N}$. Thus every object is a product of finitely many copies of the *generating object* $A := A^1$.

A *model* of a Lawvere algebraic theory $\mathbb{A}$ in any category $\mathcal{C}$ with finite products is simply an FP functor $M : \mathbb{A} \to \mathcal{C}$, and a *homomorphism of models* is just a natural transformation $\vartheta : M \to M'$ between such functors.

As was the case for the syntactic categories $\mathsf{Syn}(\mathbb{T})$, we could just as well have taken the natural numbers $0, 1, 2, \ldots$ themselves as the objects of a Lawvere algebraic theory $\mathbb{A}$, but the notation A^n is more suggestive. A Lawvere algebraic theory $\mathbb{A}$ in the sense of the above definition determines an algebraic theory in the sense of Definition 1.1.2 as follows. As basic operations with arity k we take all of the morphisms $A^k \to A$:

$$\Sigma(\mathbb{A})_k = \mathsf{Hom}_{\mathbb{A}}(A^k, A) \tag{1.18}$$

There is a canonical interpretation in $\mathbb{A}$ of terms built from variables and morphisms $A^k \to A$, namely each morphism is interpreted by itself, and variables are interpreted as product projects, as usual. An equation $u = v$ is taken as an axiom of the theory $\mathbb{A}$ just if the canonical interpretations of u and v coincide. Of course, the conventional logical notions of a model 1.1.11 and a homomorphism of models then also correspond to the new, functorial ones in the obvious way.

This more abstract view of algebraic theories immediately suggests some interesting new examples.

Example 1.2.11. The *algebraic theory of smooth maps* $\mathcal{C}^\infty$ is the category whose objects are n-dimensional Euclidean spaces $\mathsf{1}$, $\mathbb{R}$, $\mathbb{R}^2$, $\ldots$, and whose morphisms are $\mathcal{C}^\infty$-maps between them. Recall that a $\mathcal{C}^\infty$-map $f : \mathbb{R}^n \to \mathbb{R}$ is a function which has all higher partial derivatives, and that a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a $\mathcal{C}^\infty$-map exactly when all of its composites $\pi_k \circ f : \mathbb{R}^n \to \mathbb{R}$ with the projections $\pi_k : \mathbb{R}^m \to \mathbb{R}$ are $\mathcal{C}^\infty$-maps.

A model of this theory in Set is (by definition) a finite product preserving functor $A : \mathcal{C}^\infty \to \mathsf{Set}$. Up to natural isomorphism, such a model can be described as follows. A $\mathcal{C}^\infty$*-algebra* is given by a set A and for every smooth map $f : \mathbb{R}^n \to \mathbb{R}$ a function $Af : A^n \to A$ such that if $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^m \to \mathbb{R}$, $i = 1, \ldots, n$, are smooth maps then, for all $a_1, \ldots, a_m \in A$,

$$Af\big((Ag_1)\langle a_1, \ldots, a_m\rangle, \ldots, (Ag_n)\langle a_1, \ldots, a_m\rangle\big) = A(f \circ \langle g_1, \ldots, g_n\rangle)\langle a_1, \ldots, a_m\rangle \ .$$

In particular, since multiplication and addition are smooth maps, A is a commutative ring with unit. Such structures are known as *$\mathcal{C}^\infty$-rings.* Therefore, the models in Set of the theory of smooth maps are the $\mathcal{C}^\infty$-rings (cf. [MR91]).

Example 1.2.12. Recall that a *(total) recursive function* $f : \mathbb{N}^m \to \mathbb{N}^n$ is one that can be computed by a Turing machine. This means that there exists a Turing machine which on input $\langle a_1, \dots, a_m \rangle$ outputs the value of $f\langle a_1, \dots, a_m \rangle$. The algebraic theory Rec of recursive functions is the category whose objects are finite powers of the natural numbers $\mathsf{1}$, $\mathbb{N}$, $\mathbb{N}^2$, ..., and whose morphisms are recursive functions between them. The models of this theory in a category $\mathcal{C}$ with finite products give a notion of computability in $\mathcal{C}$.

Let us consider the category of all set-theoretic models $\mathcal{R} = \mathsf{Mod}(\mathsf{Rec})$. First, there is the "identity" model $I \in \mathcal{R}$, defined by $I\mathbb{N}^k = \mathbb{N}^k$ and $If = f$. Given any model $S \in \mathcal{R}$, its object part is determined by $S_1 = S\mathbb{N}$ since $S\mathbb{N}^k = S_1^k$. For every $n \in \mathbb{N}$ there is a morphism $\mathsf{1} \to \mathbb{N}$ in Rec defined by $\star \mapsto n$. Thus we have for each $n \in \mathbb{N}$ an element $s_n = S(\star \mapsto n) : \mathsf{1} \to S_1$. This defines a function $s : \mathbb{N} \to S_1$ which in turn determines a natural transformation $\sigma : I \Longrightarrow S$ whose component at $\mathbb{N}^k$ is $s \times \cdots \times s : \mathbb{N}^k \to S_1^k$.

Example 1.2.13. In a category $\mathcal{C}$ with finite products every object $A \in \mathcal{C}$ determines a full subcategory consisting of the finite powers $\mathsf{1}, A, A^2, A^3, \dots$ and all morphisms between them. This is the *total theory* $\mathbb{T}_A$ of the object A in $\mathcal{C}$.

Free algebras

In order to extend the logical duality of the foregoing section to the abstract case, we will require the notion of a *free model* of an abstract algebraic theory. Of course, we already have the conventional notion of free models determined in terms of the associated conventional algebraic theory given by (1.18). But we can also determine free models directly in terms of the abstract theory, in a way which then agrees with the conventional ones.

Let $\mathbb{A}$ be a Lawvere algebraic theory, with objects $1, A, A^2, \dots$. We have the category of models,

$$\mathsf{Mod}(\mathbb{A}) = \mathsf{Hom_{FP}}(\mathbb{A}, \mathsf{Set}).$$

Let us first define the *forgetful functor* by evaluating at the generating object $A \in \mathbb{A}$,

$$U := \mathsf{eval}_A : \mathsf{Mod}(\mathbb{A}) \to \mathsf{Set} \tag{1.19}$$

$$(M : \mathbb{A} \to \mathsf{Set}) \mapsto M(A). \tag{1.20}$$

As before, we shall also write

$$|M| = U(M) = M(A). \tag{1.21}$$

Now for the *finitary* free functor $F : \mathsf{Set}_{\mathrm{fin}} \to \mathsf{Mod}(\mathbb{A})$, we set:

$$\begin{aligned} F(0) &= \mathsf{Hom}_{\mathbb{A}}(1, -) \\ F(1) &= \mathsf{Hom}_{\mathbb{A}}(A, -) \\ &\vdots \\ F(n) &= \mathsf{Hom}_{\mathbb{A}}(A^n, -). \end{aligned}$$

Note that this is a composite of the two (contravariant) functors,

$$\mathsf{Set}_{\mathrm{fin}} \to \mathbb{A}^{\mathsf{op}} \to \mathsf{Mod}(\mathbb{A}),$$

given by $n \mapsto A^n$ and $X \mapsto \mathsf{Hom}_{\mathbb{A}}(X, -)$, and is therefore (covariantly) functorial. Note also that the representables $\mathsf{Hom}_{\mathbb{A}}(A^n, -)$ do indeed preserve finite products, and are therefore in the full subcategory $\mathsf{Mod}(\mathbb{A}) \hookrightarrow \mathsf{Set}^{\mathbb{A}}$ of models.

For adjointness we need to check that for any FP-functor $M : \mathbb{A} \to \mathsf{Set}$ there is a natural (in both arguments) bijection,

$$\mathsf{Hom}_{\mathsf{Mod}(\mathbb{A})}(F(n), M) \cong \mathsf{Hom}_{\mathsf{Set}}(n, U(M))\,. \tag{1.22}$$

The right-hand side is plainly just $|M|^n$. For the left-hand side we have:

$$\begin{aligned} \mathsf{Hom}_{\mathsf{Mod}(\mathbb{A})}(F(n), M) &= \mathsf{Hom}_{\mathsf{Mod}(\mathbb{A})}(\mathsf{Hom}_{\mathbb{A}}(A^n, -), M) \\ &= \mathsf{Hom}_{\mathsf{Set}^{\mathbb{A}}}(\mathsf{Hom}_{\mathbb{A}}(A^n, -), M) && (\mathsf{Mod}(\mathbb{A}) \text{ is full}) \\ &\cong M(A^n) && (\text{by Yoneda}) \\ &\cong M(A)^n && (M \text{ is FP}) \\ &= |M|^n && (1.21). \end{aligned}$$

The full definition of the free functor

$$F : \mathsf{Set} \to \mathsf{Mod}(\mathbb{A})$$

is then given by writing an arbitrary set X as a (filtered) colimit of its finite subsets $X_i \subseteq X$, and setting $F(X) = \mathrm{colim}_i\, F(X_i)$ in the category $\mathsf{Set}^{\mathbb{A}}$. Since filtered colimits commute with finite products, these colimits taken in $\mathsf{Set}^{\mathbb{A}}$ will remain in $\mathsf{Mod}(\mathbb{A})$.

Theorem 1.2.14. *For any set X with free algebra $F(X)$ as just defined, and any $\mathbb{A}$-model M, there is a natural isomorphism,*

$$\mathsf{Hom}_{\mathsf{Mod}(\mathbb{A})}(F(X), M) \cong \mathsf{Hom}_{\mathsf{Set}}(X, U(M))\,. \tag{1.23}$$

Proof. The rest of the proof is now an easy exercise. □

By definition, the finitely generated free models $F(n)$ are just the representables $\mathsf{Hom}_{\mathbb{A}}(A^n, -)$; therefore as the "semantic dual" $\mathsf{mod}(\mathbb{A}) \hookrightarrow \mathsf{Mod}(\mathbb{A})$ of the theory $\mathbb{A}$, in the sense of corollary 1.2.3, we simply have (the full subcategory of $\mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set})$ on) the image of the Yoneda embedding,

$$\begin{tikzcd}
\mathsf{mod}(\mathbb{A}) \arrow[r, hook] & \mathsf{Mod}(\mathbb{A}) = \mathsf{Hom}_{\mathrm{FP}}(\mathbb{A}, \mathsf{Set}) \arrow[r, hook] & \mathsf{Set}^{\mathbb{A}} \\
\mathbb{A}^{\mathsf{op}} \arrow[u, "\simeq"] \arrow[rr, "="'] & & \mathbb{A}^{\mathsf{op}}. \arrow[u, hook, "\mathsf{y}"']
\end{tikzcd}$$

So in the abstract case, the logical duality

$$\mathbb{A} \simeq \mathsf{mod}(\mathbb{A})^{\mathsf{op}}$$

comes down to the fact that the (contravariant) Yoneda embedding

$$\mathbb{A}^{\mathsf{op}} \hookrightarrow \mathsf{Set}^{\mathbb{A}}$$

represents $\mathbb{A}$ as (the dual of) a full subcategory of (product-preserving!) functors.

Summarizing, we have now shown:

Theorem 1.2.15. *For any Lawvere algebraic theory $\mathbb{A}$, there is an equivalence,*

$$\mathbb{A} \simeq \mathsf{mod}(\mathbb{A})^{\mathsf{op}}$$

between $\mathbb{A}$ and the full subcategory of finitely generated free models.

Exercise 1.2.16. Prove theorem 1.2.14.

1.2.3 Algebraic categories

Given an *arbitrary* category $\mathcal{A}$, we may ask: *When is $\mathcal{A}$ the category of models for some algebraic theory?* Such categories are sometimes called *varieties*, at least in universal algebra, and there are well-known *recognition theorems* such as Birkhoff's famous HSP-theorem, which says that a class of interpretations for some fixed signature are all those satisfying a set of equations if the class is closed under Products, Subalgebras, and Homomorphic images (i.e. quotients by an algebra congruence). Toward the goal of "recognizing" a *category* of algebras (without being given the signature!), let us define:

Definition 1.2.17. An *algebraic category $\mathcal{A}$* is a (locally small) category equivalent to one of the form

$$\mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set}) \hookrightarrow \mathsf{Set}^{\mathbb{A}}$$

where $\mathbb{A}$ is any small finite product category and $\mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set})$ is the full subcategory of finite product preserving functors. If $\mathbb{A}$ is a Lawvere algebraic theory (i.e. the objects are generated under finite products by a single object), then we will say that $\mathcal{A}$ is a *Lawvere algebraic category.*

If $\mathcal{A} \simeq \mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set})$ is a Lawvere algebraic category, then in particular there will be a forgetful functor, determined by evaluation at the generating object A of $\mathbb{A}$,

$$U = \mathsf{eval}_A : \mathcal{A} \to \mathsf{Set}. \tag{1.24}$$

It follows immediately that U preserves all limits, and one can show without difficulty that it also preserves all filtered colimits (cf. exercise 1.2.23). We require only one further condition to "recognize" $\mathcal{A}$ as algebraic, namely creation of "U-absolute coequalizers".

Definition 1.2.18. In any category $\mathcal{D}$, a coequalizer $c : Y \to Z$ of maps $a, b : X \rightrightarrows Y$ is *absolute* if, for every category $\mathcal{D}'$ and functor $G : \mathcal{D} \to \mathcal{D}'$, the map $Gc : GY \to GZ$ is a coequalizer of the maps $Ga, Gb : GX \rightrightarrows GY$. A functor $F : \mathcal{C} \to \mathcal{D}$ may be said to *create F-absolute coequalizers* if for every parallel pair of maps $a, b : X \rightrightarrows Y$ in $\mathcal{C}$ and absolute coequalizer $q : FY \to Q$ of $Fa, Fb : FX \rightrightarrows FY$ in $\mathcal{D}$, there is a unique object Z and map $c : Y \to Z$ in $\mathcal{C}$ with $FZ = Q$ and $Fc = q$, which, moreover, is a coequalizer in $\mathcal{C}$.

$$\begin{tikzcd}
\mathcal{C} \arrow[d, "F"'] & X \arrow[r, shift left, "a"] \arrow[r, shift right, "b"'] & Y \arrow[r, dotted, "c"] & Z \\
\mathcal{D} & FX \arrow[r, shift left, "Fa"] \arrow[r, shift right, "Fb"'] & FY \arrow[r, "q"'] & Q
\end{tikzcd} \tag{1.25}$$

Thus, roughly, F creates those coequalizers in $\mathcal{C}$ that are absolute in $\mathcal{D}$.

Theorem 1.2.19. *Given a category $\mathcal{A}$ equipped with a functor $U : \mathcal{A} \to \mathsf{Set}$, the following conditions are equivalent.*

1. *$\mathcal{A}$ is a Lawvere algebraic category, and $U \cong \mathsf{eval}_A : \mathcal{A} \to \mathsf{Set}$. In more detail, there is a Lawvere algebraic theory $\mathbb{A}$, and an equivalence,*

$$\mathcal{A} \simeq \mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set}) \hookrightarrow \mathsf{Set}^{\mathbb{A}},$$

associating $U : \mathcal{A} \to \mathsf{Set}$ to the evaluation at the generating object of $A \in \mathbb{A}$.

2. *$U : \mathcal{A} \to \mathsf{Set}$ has a left adjoint $F : \mathsf{Set} \to \mathcal{A}$, preserves all filtered colimits, and creates U-absolute coequalizers.*

3. *$\mathcal{A}$ is monadic over Set (via $U : \mathcal{A} \to \mathsf{Set}$),*

$$\mathcal{A} \simeq \mathsf{Set}^T$$

for a finitary monad $T : \mathsf{Set} \to \mathsf{Set}$.

Proof. (1⇒2) Suppose first that $\mathcal{A}$ is Lawvere algebraic, so

$$\mathcal{A} \simeq \mathsf{Mod}(\mathbb{A}) = \mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set}) \hookrightarrow \mathsf{Set}^{\mathbb{A}}$$

for a Lawvere algebraic theory $\mathbb{A}$. By theorem 1.2.14 we know that U has a left adjoint $F : \mathsf{Set} \to \mathcal{A}$. It also preserves filtered colimits because they commute with finite products, and so a filtered colimit of FP functors, calculated in $\mathsf{Set}^{\mathbb{A}}$, is again an FP functor. This suffices, since colimits are computed pointwise in $\mathsf{Set}^{\mathbb{A}}$ and U is an evaluation functor.

For creation of U-absolute coequalizers, suppose we have maps $f, g : A \rightrightarrows B$ in $\mathcal{A}$ and an absolute coequalizer $c : UB \to C$ for $Uf, Ug : UA \rightrightarrows UB$ in Set; we want to put an algebra structure on C making c a homomorphism $c : B \to C$, and a coequalizer of f and g in $\mathcal{A}$.

$$\begin{tikzcd}
UA^n \arrow[r, shift left, "Uf^n"] \arrow[r, shift right, "Ug^n"'] \arrow[d, "\sigma^A"'] & UB^n \arrow[r, "c^n"] \arrow[d, "\sigma^B"] & C^n \arrow[d, dotted, "\sigma^C"] \\
UA \arrow[r, shift left, "Uf"] \arrow[r, shift right, "Ug"'] & UB \arrow[r, "c"'] & C
\end{tikzcd} \tag{1.26}$$

For each function symbol $\sigma \in \Sigma$ we have commutative squares as on the left in the above diagram, because f and g are homomorphisms. It follows by a simple diagram chase that $c \circ \sigma^B$ coequalizes the pair $Uf^n, Ug^n : UA^n \rightrightarrows UB^n$. Since $c : UB \to C$ is absolute, it is preserved by the functor $(-)^n$, and therefore $c^n : UB^n \to C^n$ is a coequalizer of Uf^n, Ug^n. There is therefore a unique map $\sigma^C : C^n \to C$ as indicated, making the right hand square commute. Doing this for each $\sigma \in \Sigma$ gives an interpretation of Σ on C. This is seen to be an algebra structure because the maps c^n are surjections. Thus $c : B \to C$ becomes a homomorphism, which is easily seen to be a coequalizer in $\mathcal{A}$.

(2⇒3) Taking the standard monad (T, η, μ) on Set with underlying functor $T = U \circ F$, we want to show that the canonical comparison map

$$\mathcal{A} \to \mathsf{Set}^T$$

to the category of T-algebras is an isomorphism. This follows by Beck's theorem (see [Lan71, VI.7]) from the condition that U creates absolute coequalizers. Moreover, T preserves filtered colimits (i.e. is "finitary") because each of F and U do so.

(3⇒1) Let (T, η, μ) be a finitary monad on Set and $U : \mathsf{Set}^T \to \mathsf{Set}$ the forgetful functor from the category of T-algebras. We want an algebraic theory $\mathbb{A}$ and an equivalence

$$\mathsf{Set}^T \simeq \mathsf{Mod}(\mathbb{A})$$

commuting with U and evaluation at the generator of $\mathbb{A}$, where recall $\mathsf{Mod}(\mathbb{A}) = \mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set})$. Let

$$\mathbb{A} = (\mathsf{Set}^T)^{\mathsf{op}}_{\mathsf{fgf}} \tag{1.27}$$

be the dual of the full subcategory of finitely generated free T-algebras. The objects of $\mathbb{A}$ are thus of the form $T_0, T_1, T_2, ...$ where $T_n = T(n)$, equipped with the multiplication $\mu_n : T^2(n) \to T(n)$ as algebra structure map. Since, as free algebras, $T(n+m) \cong T(n) + T(m)$ we indeed have $T_n \times T_m \cong T_{n+m}$ as objects of $\mathbb{A}$, and T_1 as the generating object.
By the first two steps of this proof, we know that the algebraic category $\mathsf{Mod}(\mathbb{A})$ is also (finitary) monadic,

$$\mathsf{Mod}(\mathbb{A}) \simeq \mathsf{Set}^M,$$

with monad $M = U_M \circ F_M$, where $F_M \dashv U_M$ is the free-forgetful adjunction for $\mathsf{Mod}(\mathbb{A}) = \mathsf{Hom}_{\mathsf{FP}}(\mathbb{A}, \mathsf{Set})$, and $U_M \cong \mathsf{eval}_{T_1}$. Thus it will suffice to show that $M \cong T$, as monads, in order to conclude that

$$\mathsf{Mod}(\mathbb{A}) \simeq \mathsf{Set}^M \simeq \mathsf{Set}^T.$$

Moreover, since both M and T are finitary, it suffices to show that their respective restrictions to the dense subcategory $\mathsf{Set}_{\mathsf{fin}} \hookrightarrow \mathsf{Set}$ are isomorphic. By (1.22), we know that the finite free functor $F_M(n)$ has the form

$$F_M(n) = \mathsf{Hom}_{\mathbb{A}}(T_n, -) = \mathsf{Hom}_{(\mathsf{Set}^T)_{\mathsf{fgf}}}(-, \langle T(n), \mu_n \rangle)$$

thus using the fact that $U_M \cong \mathsf{eval}_{T_1}$ we see that

$$\begin{aligned} M(n) = U_M(F_M(n)) &= U_M\big(\mathsf{Hom}_{(\mathsf{Set}^T)_{\mathsf{fgf}}}(-, \langle T(n), \mu_n \rangle)\big) \\ &\cong \mathsf{Hom}_{(\mathsf{Set}^T)_{\mathsf{fgf}}}(\langle T(1), \mu_1 \rangle, \langle T(n), \mu_n \rangle) \\ &\cong \mathsf{Hom}_{\mathsf{Set}}(1, T(n)) \cong T(n). \end{aligned}$$

□

This theorem can be used to show, for example, that the theory of posets cannot be given an equational reformulation (unlike $\wedge$-semilattices, which can), by showing that the category Pos is not Lawvere algebraic (via the underlying set functor $U : \mathsf{Pos} \to \mathsf{Set}$).

Exercise 1.2.20. Show this. (*Hint:* determine the left adjoint of U, and the resulting monad $U \circ F : \mathsf{Set} \to \mathsf{Set}$.)

Remark 1.2.21. Another "recognition theorem" that can be found in [Bor94] is the following:

Theorem (Borceux II.3.9)**.** *Given a category $\mathcal{A}$, equipped with a functor $U : \mathcal{A} \to \mathsf{Set}$, the following conditions are equivalent.*

1. *$\mathcal{A}$ is equivalent to the category of models of some Lawvere algebraic theory $\mathbb{T}$,*

$$\mathcal{A} \simeq \mathsf{Mod}(\mathbb{T})$$

with $U : \mathcal{A} \to \mathsf{Set}$ the corresponding forgetful functor.

2. *$\mathcal{A}$ has coequalizers and kernel pairs, and $U : \mathcal{A} \to \mathsf{Set}$ has a left adjoint $F : \mathsf{Set} \to \mathcal{A}$, preserves all filtered colimits and regular epimorphisms, and reflects isomorphisms.*

Exercise 1.2.22. A *split coequalizer* for maps $f, g : A \rightrightarrows B$ is a map $e : B \to C$ together with s and t as indicated below,

$$\begin{tikzcd}
A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "e"] \arrow[l, bend left, "t"] & C \arrow[l, bend left, "s"]
\end{tikzcd} \tag{1.28}$$

satisfying the equations

$$ef = eg, \quad ft = 1_B, \quad gt = se, \quad es = 1_C.$$

Show that a split coequalizer is an absolute coequalizer.

Exercise 1.2.23. A filtered colimit of algebras can be described directly as follows: First consider the case of sets. Let the index category $\mathbb{J}$ be filtered and $D : \mathbb{J} \to \mathsf{Set}$ a diagram. The colimiting set $\mathrm{colim}_j\, D_j$ can be described as the quotient of the coproduct $(\coprod_j Dj)/\!\sim$, where the equivalence relation $\sim$ is defined by:

$$(d_i \in D_i) \sim (d_j \in D_j) \;\Leftrightarrow\; t_{ik}(d_i) = t_{jk}(d_j) \text{ for some } t_{ik} : i \to k \text{ and } t_{jk} : j \to k \text{ in } \mathbb{J}.$$

1. Show that this is an equivalence relation using the filteredness of $\mathbb{J}$.
2. Now assume that the D_j all have an algebra structure and that all the transition maps $t_{ik} : D_i \to D_k$ are homomorphisms. Show that the colimit set $D_\infty = \mathrm{colim}_j\, D_j$ is also an algebra of the same kind by defining each of the operations $\sigma_\infty : D_\infty \times ... \times D_\infty \to D_\infty$ on equivalence classes as

   $$\sigma_\infty\langle [d_i], ..., [d_j']\rangle = [\sigma_k\langle t_{ik}(d_i), ..., t_{jk}(d_j')\rangle]$$

   for suitable k. Show that this is well-defined, and that D_∞, so equipped, also satisfies the equations satisfied by the D_j.

Example 1.2.24. A field is a ring in which every non-zero element has a multiplicative inverse. The theory of fields is (apparently) not algebraic, because the axiom

$$x \neq 0 \;\Rightarrow\; \exists y(x \cdot y = 1)$$

is not simply an equation. But in principle there could be an equivalent algebraic formulation of the theory which would somehow circumvent this problem. We can show that this is not the case by proving that the category Field of fields and field homomorphisms is not algebraic.

First observe that a category of models $\mathsf{Mod}(\mathbb{A})$ always has a terminal object because Set has a terminal object $\mathsf{1}$, and the constant functor $\Delta_{\mathsf{1}} : \mathbb{A} \to \mathsf{Set}$ which maps everything to $\mathsf{1}$ is a model. The functor Δ_{1} is the terminal object in $\mathsf{Mod}(\mathbb{A})$ because it is the terminal functor in the functor category $\mathsf{Set}^{\mathbb{A}}$. In order to see that Field is not algebraic it thus suffices to show that there is no terminal field.

Exercise 1.2.25. Show that the category Field does not have a terminal object. (Hint: suppose that T is the terminal field and use the unique homomorphism $\mathbb{Z}_2 \to T$ to see that $1+1=0$ in T, then reason similarly using the unique homomorphism $\mathbb{Z}_3 \to T$.)

1.2.4 Algebraic functors

Now that we know by Theorem 1.2.19 what the algebraic categories are, we would like to know what the "algebraic functors" between them are. These will be the functors induced by "syntactic translations" between theories, in the following sense. Classically, a *syntactic translation* of one algebraic theory into another may be defined as an assignment of types to types and terms to terms, respecting the tupling operations and substitutions of terms for variables. Such a translation can of course be described abstractly as a finite product preserving functor,

$$T : \mathbb{A} \to \mathbb{B}$$

between the associated (Lawvere) algebraic theories. Every such translation then induces a functor on the semantics, just by precomposition:

$$T^*(M) = M \circ T.$$

$$\begin{tikzcd}
\mathsf{Mod}(\mathbb{A}) & \mathsf{Mod}(\mathbb{B}) \arrow[l, "T^*"']
\end{tikzcd} \tag{1.29}$$

$$\begin{tikzcd}
\mathbb{A} \arrow[r, "T"] \arrow[dr, "T^*(M)"'] & \mathbb{B} \arrow[d, "M"] \\
& \mathsf{Set}
\end{tikzcd}$$

Such a functor may be regarded as being "definable" by the translation.

For instance, let $\mathbb{A}_0 = (\mathsf{Set}_{\mathrm{fin}})^{\mathsf{op}}$ be (the classifying category of) the trivial theory $\mathbb{T}_0$ of an object (so $\mathbb{A}_0 \simeq \mathcal{C}_{\mathbb{T}_0}$), so that $\mathsf{Mod}(\mathbb{A}_0) \simeq \mathsf{Set}$. Then for any Lawvere algebraic theory $\mathbb{A}$, the generating object $A \in \mathbb{A}$ has a classifying functor

$$A : \mathbb{A}_0 \to \mathbb{A}$$

which induces the forgetful functor by precomposition:

$$\begin{tikzcd}
\mathsf{Set} \simeq \mathsf{Mod}(\mathbb{A}_0) & \mathsf{Mod}(\mathbb{A}) \arrow[l, "A^*"']
\end{tikzcd}$$

$$\begin{tikzcd}
\mathbb{A}_0 \arrow[r, "A"] \arrow[dr, "A^* M"'] & \mathbb{A} \arrow[d, "M"] \\
& \mathsf{Set}
\end{tikzcd}$$

More generally, by the universal property of $\mathbb{A}$, a translation $T : \mathbb{A} \to \mathbb{B}$ corresponds to a “model of $\mathbb{A}$ in the syntax of $\mathbb{B}$”:

$$\begin{array}{c} T : \mathbb{A} \to \mathbb{B} \\ \hline\hline \hat{T} \in \mathsf{Mod}(\mathbb{A}, \mathbb{B}) \end{array}$$

For instance, since every ring R has an underlying group $|R|_{\mathsf{Grp}}$, the universal ring $\mathcal{U}_{\mathbb{R}}$ in the theory of rings $\mathbb{R}$ also has one $|\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}}$, which is therefore classified by an essentially unique functor from the theory of groups,

$$|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}} : \mathbb{G} \longrightarrow \mathbb{R},$$

which is essentially determined by taking the universal group $\mathcal{U}_{\mathbb{G}}$ to $|\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}}$:

$$|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}}(\mathcal{U}_{\mathbb{G}}) = |\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}} \,.$$

This translation induces a functor in the opposite direction on the corresponding categories of models,

$$(|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}})^* : \mathsf{Ring} \longrightarrow \mathsf{Group}, \tag{1.30}$$

taking a ring $R : \mathbb{R} \to \mathsf{Set}$ to the group,

$$(|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}})^*(R) = R \circ (|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}}) : \mathbb{G} \longrightarrow \mathsf{Set}$$

which takes the universal group $\mathcal{U}_{\mathbb{G}}$ to :

$$(R \circ (|\mathcal{U}_{\mathbb{R}}|^{\sharp}_{\mathsf{Grp}}))\mathcal{U}_{\mathbb{G}} = R(|\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}}) = |R|_{\mathsf{Grp}} \,,$$

This is of course just the underlying group functor $| - |_{\mathsf{Grp}} : \mathsf{Ring} \to \mathsf{Group}$.

More simply put, the underlying group functor $|-|_{\mathsf{Grp}} : \mathsf{Ring} \to \mathsf{Group}$ is *represented* by the translation induced by the "universal underlying group of a ring" $|\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}}$,

$$\begin{tikzcd}
\mathsf{Ring} \arrow[r, "|-|_{\mathsf{Grp}}"] \arrow[d, "\simeq"'] & \mathsf{Group} \arrow[d, "\simeq"] \\
\mathsf{Mod}(\mathbb{R}) \arrow[d, "\simeq"'] & \mathsf{Mod}(\mathbb{G}) \arrow[d, "\simeq"] \\
\mathsf{Hom}_{\mathsf{FP}}(\mathbb{R}, \mathsf{Set}) \arrow[r, "|\mathcal{U}_{\mathbb{R}}|^*_{\mathsf{Grp}}"'] & \mathsf{Hom}_{\mathsf{FP}}(\mathbb{G}, \mathsf{Set})
\end{tikzcd} \tag{1.31}$$

$$\begin{tikzcd}
\mathbb{R} \arrow[d, "R"'] & \mathbb{G} \arrow[l, "|\mathcal{U}_{\mathbb{R}}|_{\mathsf{Grp}}"'] \arrow[dl, "|R|"] \\
\mathsf{Set} &
\end{tikzcd}$$

We can now ask: *Which functors $f : \mathsf{Mod}(\mathbb{B}) \to \mathsf{Mod}(\mathbb{A})$ between algebraic categories are of the form $f = T^*$ for a translation $T : \mathbb{A} \to \mathbb{B}$ of theories?* Let us call these *algebraic functors.* We consider first the case where $\mathbb{A}$ and $\mathbb{B}$ are Lawvere algebraic and T takes the generator A of $\mathbb{A}$ to the generator B of $\mathbb{B}$,

$$T(A) \cong B$$

as in the foregoing example. Then T^* commutes with the forgetful functors, which, recall, are evaluation at the generators, $U_{\mathbb{A}}(M) = M(A)$:

$$\begin{tikzcd}
\mathsf{Mod}(\mathbb{B}) \arrow[rr, "T^*"] \arrow[dr, "U_{\mathbb{B}}"'] & & \mathsf{Mod}(\mathbb{A}) \arrow[dl, "U_{\mathbb{A}}"] \\
& \mathsf{Set} &
\end{tikzcd}$$

This is simply because

$$(U_{\mathbb{A}} \circ T^*)(M) = U_{\mathbb{A}}(M \circ T) = (M \circ T)(A) \cong M(T(A)) \cong M(B) = U_{\mathbb{B}}(M).$$

We shall see that this condition is in fact already sufficient! We first require the following.

Lemma 1.2.26. *Let $\mathcal{A}$ be Lawvere algebraic. The forgetful functor $U : \mathcal{A} \to \mathsf{Set}$ not only preserves, but also* creates *all small limits, filtered colimits, and regular epimorphisms.*

Proof. This is a standard fact, and not difficult to prove; the reader can either prove it as an exercise, or look it up in [ALR03]. □

Proposition 1.2.27. *For Lawvere algebraic theories $\mathbb{A}$ and $\mathbb{B}$, every functor $f : \mathsf{Mod}(\mathbb{B}) \to \mathsf{Mod}(\mathbb{A})$ with*

$$U_{\mathbb{B}} \cong U_{\mathbb{A}} \circ f,$$

is of the form $f \cong T^*$ for a unique (up to iso) FP-functor $T : \mathbb{A} \to \mathbb{B}$.

Proof 1. Consider the diagram

$$\begin{tikzcd}
\mathsf{Mod}(\mathbb{B}) \arrow[rr, "f"] \arrow[dr, shift left, "U_{\mathbb{B}}"] & & \mathsf{Mod}(\mathbb{A}) \arrow[dl, shift right, "U_{\mathbb{A}}"'] \\
& \mathsf{Set} \arrow[ul, shift left, "F_{\mathbb{B}}"] \arrow[ur, shift right, "F_{\mathbb{A}}"'] &
\end{tikzcd}$$

where, in each pair, we have an adjunction $F \dashv U$ by Theorem 1.2.19, and the central triangle commutes up to iso. We seek an FP-functor $T : \mathbb{A} \to \mathbb{B}$ such that $f \cong T^*$.

Since by Lemma 1.2.26, $U_{\mathbb{A}}$ creates limits, and $U_{\mathbb{A}} \circ f \cong U_{\mathbb{B}}$ preserves them, it follows that f also preserves them. In more detail, given a diagram $D : J \to \mathsf{Mod}(\mathbb{B})$ with limit $\lim_j D_j$, we have $f \lim_j D_j \cong \lim_j f D_j$ just if $U_{\mathbb{A}} f \lim_j D_j \cong \lim_j U_{\mathbb{A}} f D_j$, since $U_{\mathbb{A}}$ creates limits. But

$$U_{\mathbb{A}} f \lim_j D_j \cong U_{\mathbb{B}} \lim_j D_j \cong \lim_j U_{\mathbb{B}} D_j \cong \lim_j U_{\mathbb{A}} f D_j$$

since $U_{\mathbb{A}} f \cong U_{\mathbb{B}}$ and $U_{\mathbb{B}}$ preserves limits. The same argument applies to filtered colimits (and regular epis).

Now both $\mathsf{Mod}(\mathbb{A})$ and $\mathsf{Mod}(\mathbb{B})$ are *locally finitely presentable* (LFP): a cocomplete category $\mathcal{C}$ is LFP if it has a small subcategory $\mathbb{K}$ of finitely presentable objects k such that every object c in $\mathcal{C}$ is a filtered colimit of all the maps $k \to c$.[1] A category of algebras like $\mathsf{Mod}(\mathbb{A})$ is LFP because it is a reflective subcategory of a functor category $\mathsf{Set}^{\mathbb{A}}$, with a filtered-colimit preserving inclusion (cf. [AR94]). Thus, since $f : \mathsf{Mod}(\mathbb{B}) \to \mathsf{Mod}(\mathbb{A})$ preserves (small) limits and filtered colimits, it therefore has a left adjoint

$$f_! : \mathsf{Mod}(\mathbb{A}) \to \mathsf{Mod}(\mathbb{B})$$

by the Adjoint Functor Theorem. Indeed, one can check the solution set condition directly (see [Lan71, AR94]). From $U_{\mathbb{B}} \cong U_{\mathbb{A}} \circ f$, it then follows that $F_{\mathbb{B}} \cong f_! \circ F_{\mathbb{A}}$. In particular, for the generators $A = F_{\mathbb{A}}(1)$ and $B = F_{\mathbb{B}}(1)$ we have $f_!(A) = f_! F_{\mathbb{A}}(1) \cong F_{\mathbb{B}}(1) = B$, and then $f_!(\coprod_n A) = \coprod_n B$ since $f_!$ preserves coproducts.

Since we know by Theorem 1.2.15 that $\mathbb{A} \simeq \mathsf{mod}(\mathbb{A})^{\mathsf{op}}$, the dual of the subcategory of finitely generated free models, and the same holds for $\mathbb{B}$, the left adjoint $f_! : \mathsf{Mod}(\mathbb{A}) \to$

[1] An object k is called *finitely presentable* if $\mathsf{Hom}(k, -)$ preserves filtered colimits, see [AR94]

$\mathsf{Mod}(\mathbb{B})$ restricts and dualizes to an FP "translation of theories" $T : \mathbb{A} \to \mathbb{B}$ as in,

$$\begin{tikzcd}
\mathbb{A} \arrow[r, dotted, "T"] \arrow[d, hook, "\mathsf{y}^{\mathsf{op}}"'] & \mathbb{B} \arrow[d, hook, "\mathsf{y}^{\mathsf{op}}"] \\
\mathsf{Mod}(\mathbb{A})^{\mathsf{op}} \arrow[r, "f_!^{\mathsf{op}}"'] & \mathsf{Mod}(\mathbb{B})^{\mathsf{op}}
\end{tikzcd}$$

such that

$$T(A^n) = T(F_{\mathbb{A}}(n)) = f_!^{\mathsf{op}}(F_{\mathbb{A}}(n)) \cong F_{\mathbb{B}}(n) = B^n. \tag{1.32}$$

It remains to see that $f \cong T^* : \mathsf{Mod}(\mathbb{B}) \to \mathsf{Mod}(\mathbb{A})$. Indeed, for any model $M : \mathbb{B} \to \mathsf{Set}$, we have

$$\begin{aligned}
f(M)(A) &\cong \mathsf{Set}^{\mathbb{A}}(\mathsf{y}A, f(M)) && \text{Yoneda} \\
&\cong \mathsf{Mod}(\mathbb{A})(\mathsf{y}A, f(M)) && \mathsf{Mod}(\mathbb{A}) \hookrightarrow \mathsf{Set}^{\mathbb{A}} \\
&\cong \mathsf{Mod}(\mathbb{B})(f_!(\mathsf{y}A), M) && f_! \dashv f \\
&\cong \mathsf{Mod}(\mathbb{B})(\mathsf{y}(TA), M) && (1.32) \\
&\cong \mathsf{Set}^{\mathbb{B}}(\mathsf{y}(TA), M) && \mathsf{Mod}(\mathbb{A}) \hookrightarrow \mathsf{Set}^{\mathbb{A}} \\
&\cong M(TA) && \text{Yoneda} \\
&\cong T^*(M)(A),
\end{aligned}$$

naturally in M. The case of an arbitrary object $A^n \in \mathbb{A}$ follows, since the models $f(M)$ and $T^*(M)$ preserve products. □

Corollary 1.2.28. *For a functor $f : \mathsf{Mod}(\mathbb{B}) \to \mathsf{Mod}(\mathbb{A})$ between Lawvere algebraic categories, the following are equivalent.*

1. *f commutes with the forgetful functors, $U_{\mathbb{A}} \circ f \cong U_{\mathbb{B}}$.*
2. *f is algebraic: $f = T^*$, for an FP functor $T : \mathbb{A} \to \mathbb{B}$ that preserves the generator, $T(A) \cong B$.*

□

The corollary tells us that functors between the (semantic) categories of algebraic structures that respect the underlying sets correspond to translations of their syntactic presentations. In fact, even more is true: there is a *bi*equivalence of categories

$$\mathsf{LAlgCat}/\mathsf{Set} \;\simeq\; \left(\mathsf{Set}_{\mathrm{fin}}^{\mathsf{op}}/\mathsf{LAlgTh}\right)^{\mathsf{op}} \;\simeq\; \left(\mathsf{LAlgTh}_{\bullet}\right)^{\mathsf{op}}, \tag{1.33}$$

where on the left we have the category of Lawvere algebraic categories $\mathcal{A}$, equipped with their canonical forgetful functors $U_{\mathbb{A}} : \mathcal{A} \to \mathsf{Set}$, and algebraic functors between them that commute up to natural isomorphism over the base, and on the right (the dual of) the category of Lawvere algebraic theories and FP functors that preserve the generator.

A “biequivalence” is like an equivalence, but only up to equivalence! Observe that the left-hand side of (1.33) is not even locally small, while the entire category on the right is small. (See e.g. [ARV10] for details.) This “global” Syntax-Semantics duality can be extended even further, as sketched in the following exercises.

Exercise 1.2.29. Show that for any Lawvere algebraic theory $\mathbb{A}$, the full inclusion $\mathsf{Mod}(\mathbb{A}) \hookrightarrow \mathsf{Set}^{\mathbb{A}}$ has a left adjoint. (*Hint:* use the Adjoint Functor Theorem.)

Exercise 1.2.30. Assuming the result of the previous exercise, show that the precomposition functor $T^* : \mathsf{Mod}(\mathbb{B}) \longrightarrow \mathsf{Mod}(\mathbb{A})$ induced by *any* translation $T : \mathbb{A} \to \mathbb{B}$ (not necessarily preserving the generator) always has a left adjoint $T_! : \mathsf{Mod}(\mathbb{A}) \longrightarrow \mathsf{Mod}(\mathbb{B})$.

Exercise 1.2.31. Assuming the results of the previous two exercises, show that an algebraic functor $f : \mathsf{Mod}(\mathbb{B}) \longrightarrow \mathsf{Mod}(\mathbb{A})$, induced by a translation $T : \mathbb{A} \to \mathbb{B}$ as $f = T^*$, satisfies the following conditions:

(i) f preserves limits.

(ii) f preserves filtered colimits.

(iii) f preserves regular epimorphisms.

Hint: Since f has a left adjoint $f_! : \mathsf{Mod}(\mathbb{A}) \longrightarrow \mathsf{Mod}(\mathbb{B})$, we know that $f_!(A) \cong B^n$ for some $0 \leq n$. Now use $\mathbb{B} \simeq \mathsf{mod}(\mathbb{B})^{\mathsf{op}}$.

Remark 1.2.32. The converse of Exercise 1.2.31 holds as well, under a certain condition on the syntactic categories, to be explained below. We then obtain a duality of the form

$$\mathsf{LAlgCat} \;\simeq\; \mathsf{LAlgTh}^{\mathsf{op}}\,, \tag{1.34}$$

generalizing (1.33) by eliminating the “base point”. This generalizes even further to “many-sorted” algebraic theories $\mathbb{A}$ not assumed to be generated by a single object, and thus given simply by small FP categories. The corresonding semantic category $\mathsf{Mod}(\mathbb{A})$ still consists of all FP-functors $\mathbb{A} \to \mathsf{Set}$, and is still locally finitely presentable. The question, when is a “semantic functor”

$$f : \mathsf{Mod}(\mathbb{B}) \longrightarrow \mathsf{Mod}(\mathbb{A})$$

between such algebraic categories induced by a “syntactic translation” $T : \mathbb{A} \to \mathbb{B}$ of such algebraic theories can also be answered in this more general setting, determining a general notion of an *algebraic functor*: it is again one that preserves all limits, filtered colimits, and regular epimorphisms. The resulting duality

$$\mathsf{AlgCat} \;\simeq\; \mathsf{AlgTh}^{\mathsf{op}}\,, \tag{1.35}$$

requires a technical condition on the syntactic side, however; namely, that the algebraic theories are closed under retracts—as does (1.34). See [ALR03] for details.

1.2.5 Dualities for algebraic theories

Let us now summarize, and generalize, the different dualities for algebraic theories that arose in this chapter. See the references [ALR03, ARV10] for details.

In Section 1.2.1, we had what we called the *logical duality* relating an individual algebraic theory $\mathbb{T}$ and its category of models $\mathsf{Mod}(\mathbb{T})$, given by an equivalence of categories

$$\mathcal{C}_{\mathbb{T}}{}^{\mathsf{op}} \simeq \mathsf{mod}(\mathbb{T}) \hookrightarrow \mathsf{Mod}(\mathbb{T}),$$

between the classifying category category $\mathcal{C}_{\mathbb{T}}$ – which can be constructed from the syntax of $\mathbb{T}$ – and a full subcategory of the semantics $\mathsf{mod}(\mathbb{T}) \hookrightarrow \mathsf{Mod}(\mathbb{T})$, consisting of the finitely generated free models.

The “semantics” functor Mod is represented by assigning to each model M an essentially unique FP functor $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathsf{Set}$, providing an equivalence of categories,

$$\mathsf{Mod}(\mathbb{T}) \;\simeq\; \mathsf{Hom}_{\mathsf{FP}}(\mathcal{C}_{\mathbb{T}}, \mathsf{Set}). \tag{1.36}$$

By (1.36), the assignment $\mathbb{T} \mapsto \mathsf{Mod}(\mathbb{T})$ is then *contravariantly functorial* in the theory $\mathbb{T}$ when we regard theories abstractly as (small) finite product categories $\mathbb{A}$, and syntactic translations as FP functors $T : \mathbb{A} \to \mathbb{B}$. This provides a global semantic representation of the syntax of algebraic theories,

$$\mathsf{Mod} : \mathsf{AlgTh}^{\mathsf{op}} \to \mathsf{Cat}\,.$$

We can recognize the essential image of the semantics functor Mod as consisting of certain *locally finitely presentable* categories $\mathcal{A} = \mathsf{Mod}(\mathbb{A})$, which may be called *algebraic categories.*[2]

An algebraic theory $\mathbb{A}$ can then be recovered functorially from its algebraic category of models $\mathcal{A} = \mathsf{Mod}(\mathbb{A})$ as the finitely generated free models. As remarked earlier, these can now also be determined “intrinsically”, by considering the category of all *algebraic functors*

$$f : \mathcal{A} \to \mathsf{Set}\,,$$

defined as functors preserving limits, filtered colimits, and regular epimorphisms. Indeed, this follows from the duality

$$\mathsf{AlgTh}^{\mathsf{op}} \;\simeq\; \mathsf{AlgCat}\,, \tag{1.37}$$

by taking $\mathsf{Set} = \mathsf{Mod}(\mathbb{A}_0)$, where $\mathbb{A}_0 = \mathsf{Set}_{\mathsf{fin}}^{\mathsf{op}}$ is the free FP category on one object (which we recognized in Example 1.2.4 as the classifying category $\mathcal{C}_{\mathbb{T}_0}$ of the trivial theory $\mathbb{T}_0$), so that we have a sequence of *equivalences*:

$$\begin{array}{cc}
\mathcal{A} \longrightarrow \mathsf{Set} & \mathsf{AlgCat} \\
\hline\hline
\mathsf{Mod}(\mathbb{A}) \longrightarrow \mathsf{Mod}(\mathbb{A}_0) & \mathsf{AlgCat} \\
\hline\hline
\mathbb{A}_0 \longrightarrow \mathbb{A} & \mathsf{AlgTh} \\
\hline\hline
\multicolumn{2}{c}{\mathbb{A}}
\end{array}$$

[2] These can be characterized as cocomplete categories $\mathcal{A}$ having a set G of objects such that every $A \in \mathcal{A}$ is a *sifted* colimit of objects $G \in \mathsf{G}$, for each of which $\mathsf{Hom}(G, -)$ preserves sifted colimits. See [ARV10].

In this way, the category Set serves as a “dualizing object”, representing both the semantics Mod and the syntax Thy as contrvariant functors,

$$\begin{tikzcd}\mathsf{AlgTh}^{\mathsf{op}} \arrow[r, bend left, "\mathsf{Mod}"] & \mathsf{AlgCat} \arrow[l, bend left, "\mathsf{Thy}"]\end{tikzcd} \tag{1.38}$$

using the two different structures on Set,

$$\begin{aligned}\mathsf{Mod}(\mathbb{A}) &\simeq \mathsf{Hom}_{\mathsf{AlgTh}}(\mathbb{A}, \mathsf{Set}),\\ \mathsf{Thy}(\mathcal{A}) &\simeq \mathsf{Hom}_{\mathsf{AlgCat}}(\mathcal{A}, \mathsf{Set}).\end{aligned}$$

The functors in (1.38) do *not* yet form a biequivalence, however, but only an adjunction. The syntax-semantics duality of (1.37) results by cutting down the syntax side AlgTh to those theories in the image of the Thy functor. These can be described as those $\mathbb{A}$ for which the unit of the adjunction

$$\eta : \mathbb{A} \longrightarrow \mathsf{Thy}(\mathsf{Mod}(\mathbb{A}))$$

is an equivalence. One can show that this holds just if the finite product category $\mathbb{A}$ is closed under retracts, and that this is always the case for $\mathsf{Thy}(\mathsf{Mod}(\mathbb{A}))$, which is then the so-called *Cauchy completion* of $\mathbb{A}$. See [ARV10].

1.2.6 Definability*

Suppose we have a (conventional, single-sorted) algebraic theory $\mathbb{T}$. Then a term in context $(\, x_1, ..., x_n \mid t\,)$ determines, for each $\mathbb{T}$-algebra A (in Set), an operation

$$t^A : A^n \to A\,,$$

that commutes with every homomorphism $h : A \to B$,

$$\begin{tikzcd}A^n \arrow[r, "h^n"] \arrow[d, "t^A"'] & B^n \arrow[d, "t^B"] \\ A \arrow[r, "h"'] & B.\end{tikzcd} \tag{1.39}$$

Suppose we are just given a family of operations $(f_A : A^n \to A)_{A \in \mathsf{Mod}(\mathbb{T})}$ commuting with all homomorphisms, in the sense of (1.39). Are they necessarily the interpretations of some term t built up from the signature $\Sigma_{\mathbb{T}}$? The answer is yes, and it’s now easy to show.

Proposition 1.2.33. *Given a family $(f_A : A^n \to A)_{A \in \mathsf{Mod}(\mathbb{T})}$ of operations on $\mathbb{T}$-algebras commuting with all homomorphisms $h : A \to B$,*

$$\begin{tikzcd}
A^n \arrow[r, "h^n"] \arrow[d, "f_A"'] & B^n \arrow[d, "f_B"] \\
A \arrow[r, "h"'] & B
\end{tikzcd} \tag{1.40}$$

there is a term in context $(\, x_1, ..., x_n \mid t\,)$ such that $f_A = t^A$ for each algebra A. Moreover, t is unique up to $\mathbb{T}$-provable equality.

Proof. Such a family is exactly a natural transformation $f : U^n \to U$ for the underlying set functor $U : \mathsf{Mod}(\mathbb{T}) \to \mathsf{Set}$. But both U and its n-fold power U^n are representable, by finitely generated free algebras, namely $U \cong \mathsf{Mod}(\mathbb{T})(F(1), -)$ and $U^n \cong \mathsf{Mod}(\mathbb{T})(F(n), -)$, and these finitely generated free algebras are objects of the classifying category $\mathcal{C}_{\mathbb{T}} \simeq \mathsf{mod}(\mathbb{T})^{\mathsf{op}}$. Since the composite functor

$$\mathsf{mod}(\mathbb{T})^{\mathsf{op}} \hookrightarrow \mathsf{Mod}(\mathbb{T})^{\mathsf{op}} \hookrightarrow \mathsf{Set}^{\mathsf{Mod}(\mathbb{T})}$$

is full and faithful, the natural transformation $f : U^n \to U$ comes from a unique arrow $F(n) \to F(1)$ in $\mathsf{mod}(\mathbb{T})^{\mathsf{op}}$, which therefore corresponds to (the equivalence class of) a term

$$(\, x_1, ..., x_n \mid t\,) : [\, x_1, ..., x_n\,] \to [\, x_1\,]$$

in the syntax of $\mathbb{T}$, which is unique up to $\mathbb{T}$-provable equality. □

A more difficult question to answer, but one to which our machinery also applies, is the following. Suppose we extend $\mathbb{T}$ to a new theory $\mathbb{T}'$ by adding a single function symbol f, together with some new equations between terms of $\mathbb{T}'$ (but not so many that new equations are implied between terms of $\mathbb{T}$). Consider the resulting FP-functor $e : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}'}$ classifying the underlying $\mathbb{T}$-model of the universal $\mathbb{T}'$-model. Precomposition with e determines a (relative) forgetful functor $E = e^* : \mathsf{Mod}(\mathbb{T}') \to \mathsf{Mod}(\mathbb{T})$. Under what conditions on the "syntactic extension" $e : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}'}$ is this functor $E : \mathsf{Mod}(\mathbb{T}') \to \mathsf{Mod}(\mathbb{T})$ full? faithful? essentially surjective? By the foregoing proposition it is all three—*i.e.* an equivalence of categories of models—if the new operation f is already *definable* by a term t in $\mathbb{T}$, in the sense that $\mathbb{T}' \vdash f(x_1, ..., x_n) = t(x_1, ..., x_n)$. The individual questions are good ones for further study. (See [Mak93] for related results.)

Chapter 2

Propositional Logic

Propositional logic is the logic of propositional connectives like $p \wedge q$ and $p \Rightarrow q$. As was the case for algebraic theories, the general approach will be to determine suitable categorical structures to model the logical operations, and then use categories with such structure to represent (abstract) propositional theories. Adjoints will play a special role, as we will describe the basic logical operations as such. We again show that the semantics is "functorial", meaning that the models of a theory are functors that preserve the categorical structure. We will show that there are classifying categories for all propositional theories, as was the case for the algebraic theories that we have already met.

A more abstract, algebraic perspective will then relate the propositional case of syntax-semantics duality with classical Stone duality for Boolean algebras, and related results from lattice theory will provide an algebraic treatment of Kripke semantics for intuitionistic (and modal) propositional logic.

2.1 Propositional calculus

Before going into the details of the categorical approach, we first briefly review the propositional calculus from a conventional point of view, as we did for algebraic theories. We focus first on *classical* propositional logic, before considering the intuitionistic case in Section 2.9.

In the style of Section B.1, we have the following (abstract) syntax for (propositional) formulas:

$$\begin{aligned} \text{Propositional variable } p &::= \mathtt{p}_1 \mid \mathtt{p}_2 \mid \mathtt{p}_3 \mid \cdots \\ \text{Propositional formula } \phi &::= p \mid \top \mid \bot \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \phi_1 \Leftrightarrow \phi_2 \end{aligned}$$

An example of a formula is therefore $\big(\mathtt{p}_3 \Leftrightarrow ((((\neg\mathtt{p}_1) \vee (\mathtt{p}_2 \wedge \bot)) \vee \mathtt{p}_1) \Rightarrow \mathtt{p}_3)\big)$. We will make use of the usual conventions for parenthesis, with binding order $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$. Thus e.g. the foregoing may also be written unambiguously as $\mathtt{p}_3 \Leftrightarrow \neg\mathtt{p}_1 \vee \mathtt{p}_2 \wedge \bot \vee \mathtt{p}_1 \Rightarrow \mathtt{p}_3$.

Natural deduction

The system of *natural deduction* for propositional logic has one form of judgement

$$\mathtt{p}_1, \dots, \mathtt{p}_n \mid \phi_1, \dots, \phi_m \vdash \phi$$

where $\mathtt{p}_1, \dots, \mathtt{p}_n$ is a *context* consisting of distinct propositional variables, the formulas $\phi_1, \dots, \phi_m$ are the *hypotheses* and ϕ is the *conclusion*. The variables in the hypotheses and the conclusion must occur among those listed in the context. The hypotheses are regarded as a (finite) set; so they are unordered, have no repetitions, and may be empty. We may abbreviate the context of variables by Γ, and we often omit it.

Deductive entailment (or *derivability*) $\Phi \vdash \phi$ is thus a relation between finite sets of formulas Φ and single formulas ϕ. It is defined as the smallest such relation satisfying the following rules:

1. Hypothesis:

$$\frac{}{\Phi \vdash \phi} \text{ if } \phi \text{ occurs in } \Phi$$

2. Truth:

$$\frac{}{\Phi \vdash \top}$$

3. Falsehood:

$$\frac{\Phi \vdash \bot}{\Phi \vdash \phi}$$

4. Conjunction:

$$\frac{\Phi \vdash \phi \qquad \Phi \vdash \psi}{\Phi \vdash \phi \wedge \psi} \qquad \frac{\Phi \vdash \phi \wedge \psi}{\Phi \vdash \phi} \qquad \frac{\Phi \vdash \phi \wedge \psi}{\Phi \vdash \psi}$$

5. Disjunction:

$$\frac{\Phi \vdash \phi}{\Phi \vdash \phi \vee \psi} \qquad \frac{\Phi \vdash \psi}{\Phi \vdash \phi \vee \psi} \qquad \frac{\Phi \vdash \phi \vee \psi \qquad \Phi, \phi \vdash \theta \qquad \Phi, \psi \vdash \theta}{\Phi \vdash \theta}$$

6. Implication:

$$\frac{\Phi, \phi \vdash \psi}{\Phi \vdash \phi \Rightarrow \psi} \qquad \frac{\Phi \vdash \phi \Rightarrow \psi \qquad \Phi \vdash \phi}{\Phi \vdash \psi}$$

For the purpose of deduction, we define $\neg\phi := \phi \Rightarrow \bot$ and $\phi \Leftrightarrow \psi := (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$. To obtain *classical* logic we need only include one of the following additional rules.

7. Classical logic:

$$\frac{}{\Phi \vdash \phi \vee \neg\phi} \qquad \frac{\Phi \vdash \neg\neg\phi}{\Phi \vdash \phi}$$

A *proof* of a judgement $\Phi \vdash \phi$ is a *finite* tree built from the above inference rules whose root is $\Phi \vdash \phi$. For example, here is a proof of $\phi \vee \psi \vdash \psi \vee \phi$ using the disjunction rules:

$$\dfrac{\dfrac{}{\phi \vee \psi \vdash \phi \vee \psi} \qquad \dfrac{\dfrac{}{\phi \vee \psi, \phi \vdash \phi}}{\phi \vee \psi, \phi \vdash \psi \vee \phi} \qquad \dfrac{\dfrac{}{\phi \vee \psi, \psi \vdash \psi}}{\phi \vee \psi, \psi \vdash \psi \vee \phi}}{\phi \vee \psi \vdash \psi \vee \phi}$$

A judgment $\Phi \vdash \phi$ is *provable* if there exists a proof of it. Observe that every proof has at its leaves either the rule for $\top$ or an instance of the rule of hypothesis (or the Excluded Middle rule for classical logic).

Remark 2.1.1. An alternate form of presentation for proofs in natural deduction that is more, well, natural uses trees of formulas, rather than of judgements, with leaves labelled by *assumptions* ϑ that may also occur in *cancelled* form $[\vartheta]$. Thus for example the introduction and elimination rules for conjunction would be written in the form:

$$\dfrac{\begin{array}{c}\Phi\\ \vdots\\ \phi\end{array} \quad \begin{array}{c}\Phi\\ \vdots\\ \psi\end{array}}{\phi \wedge \psi} \qquad\qquad \dfrac{\begin{array}{c}\Phi\\ \vdots\\ \phi \wedge \psi\end{array}}{\phi} \qquad \dfrac{\begin{array}{c}\Phi\\ \vdots\\ \phi \wedge \psi\end{array}}{\psi}$$

An example of a proof tree with cancelled assumptions is the one for disjunction elimination:

$$\dfrac{\begin{array}{c}\Phi\\ \vdots\\ \phi \vee \psi\end{array} \qquad \begin{array}{c}\Phi, [\phi]\\ \vdots\\ \vartheta\end{array} \qquad \begin{array}{c}\Phi, [\psi]\\ \vdots\\ \vartheta\end{array}}{\vartheta}$$

And the above rule of implication introduction takes the form:

$$\dfrac{\begin{array}{c}\Phi, [\phi]\\ \vdots\\ \psi\end{array}}{\phi \Rightarrow \psi}$$

In these examples, the cancellation occurred at the last step. In order to continue such a proof, we need a device to indicate *when* a cancellation occurs, *i.e.* at which step of the proof. This can be done as follows:

$$\dfrac{\dfrac{\begin{array}{c}\Phi, [\alpha]^2\\ \vdots\\ \phi \vee \psi\end{array} \qquad \begin{array}{c}\Phi, [\phi]^1\\ \vdots\\ \vartheta\end{array} \qquad \begin{array}{c}\Phi, [\psi]^1\\ \vdots\\ \vartheta\end{array}}{\vartheta}\,{\scriptstyle (1)}}{\alpha \Rightarrow \vartheta}\,{\scriptstyle (2)}$$

This proof tree represents a derivation of the judgement $\Phi \vdash \alpha \Rightarrow \vartheta$. A proof tree in which all the assumptions have been cancelled represents a derivation of an unconditional judgement such as $\vdash \phi$.

We will have a better way to record such proofs in Section **??**.

Exercise 2.1.2. Derive each of the two classical rules (2.1), called *Excluded Middle* and *Double Negation*, from the other.

2.2 Truth values

The idea of an axiomatic system of deductive, logical reasoning goes to back to Frege, who gave the first such system for propositional calculus (and more) in his *Begriffsschrift* of 1879. The question soon arose whether Frege's rules (or rather, their derivable consequences — it was clear that one could chose the primitive basis in different but equivalent ways) were correct, and if so, whether they were *all* the correct ones. An ingenious solution was proposed by Russell's student Wittgenstein, who came up with an entirely different way of singling out a set of "valid" propositional formulas in terms of assignments of truth values to the variables occurring in them. He interpreted this as showing that logical validity was really a matter of the logical structure of a proposition, rather than depending any particular system of derivations. The same idea seems to have been had independently by Post, who proved that the valid propositional formulas coincide with the ones derivable in Whitehead and Russell's *Principia Mathematica* (which is propositionally equivalent to Frege's system), a fact that we now refer to as the *soundness* and *completeness* of propositional logic.

In more detail, let a *valuation* v be an assignment of a "truth-value" $0, 1$ to each propositional variable, $v(\mathtt{p}_n) \in \{0, 1\}$. We can then extend the valuation to all propositional formulas $[\![\phi]\!]^v$ by the following recursion.

$$\begin{aligned}
[\![\mathtt{p}_n]\!]^v &= v(\mathtt{p}_n) \\
[\![\top]\!]^v &= 1 \\
[\![\bot]\!]^v &= 0 \\
[\![\neg\phi]\!]^v &= 1 - [\![\phi]\!]^v \\
[\![\phi \wedge \psi]\!]^v &= \min([\![\phi]\!]^v, [\![\psi]\!]^v) \\
[\![\phi \vee \psi]\!]^v &= \max([\![\phi]\!]^v, [\![\psi]\!]^v) \\
[\![\phi \Rightarrow \psi]\!]^v &= 1 \text{ iff } [\![\phi]\!]^v \leq [\![\psi]\!]^v \\
[\![\phi \Leftrightarrow \psi]\!]^v &= 1 \text{ iff } [\![\phi]\!]^v = [\![\psi]\!]^v
\end{aligned}$$

This is sometimes expressed using the "semantic consequence" notation $v \vDash \phi$ to mean that $[\![\phi]\!]^v = 1$. The above specification then takes the following form, in which the condition for the truth of a formula is given in terms of its informal "meaning":

$$
\begin{array}{rll}
v \vDash \top & \text{always} & \\
v \vDash \bot & \text{never} & \\
v \vDash \neg\phi & \text{iff} & \text{not } v \vDash \phi \\
v \vDash \phi \wedge \psi & \text{iff} & v \vDash \phi \text{ and } v \vDash \psi \\
v \vDash \phi \vee \psi & \text{iff} & v \vDash \phi \text{ or } v \vDash \psi \\
v \vDash \phi \Rightarrow \psi & \text{iff} & v \vDash \phi \text{ implies } v \vDash \psi \\
v \vDash \phi \Leftrightarrow \psi & \text{iff} & v \vDash \phi \text{ iff } v \vDash \psi
\end{array}
$$

Finally, ϕ is *valid*, written $\vDash \phi$, is defined by,

$$
\begin{array}{rll}
\vDash \phi & \text{iff} & v \vDash \phi \text{ for all } v \\
 & \text{iff} & [\![\phi]\!]^v = 1 \text{ for all } v\,.
\end{array}
$$

And, more generally, we define $\phi_1, ..., \phi_n$ *semantically entails* ϕ, written

$$\phi_1, ..., \phi_n \vDash \ \phi, \tag{2.1}$$

to mean that for all valuations v such that $v \vDash \phi_k$ for all k, also $v \vDash \phi$.

Given a formula in context $\Gamma \mid \phi$ and a valuation v for the variables in Γ, one can check whether $v \vDash \phi$ using a *truth table*, which is a systematic way of calculating the value of $[\![\phi]\!]^v$. For example, under the assignment $v(\mathtt{p}_1) = 1, v(\mathtt{p}_2) = 0, v(\mathtt{p}_3) = 1$ we can calculate $[\![\phi]\!]^v$ for $\phi = \big(\mathtt{p}_3 \Leftrightarrow ((((\neg\mathtt{p}_1) \vee (\mathtt{p}_2 \wedge \bot)) \vee \mathtt{p}_1) \Rightarrow \mathtt{p}_3)\big)$ as follows.

$\mathtt{p}_1$	$\mathtt{p}_2$	$\mathtt{p}_3$	$\mathtt{p}_3$	$\Leftrightarrow$	$\neg$	$\mathtt{p}_1$	$\vee$	$\mathtt{p}_2$	$\wedge$	$\bot$	$\vee$	$\mathtt{p}_1$	$\Rightarrow$	$\mathtt{p}_3$
1	0	1	1	1	0	1	0	0	0	0	1	1	1	1

The value of the formula ϕ under the valuation v is then the value in the column under the main connective, in this case $\Leftrightarrow$, and thus $[\![\phi]\!]^v = 1$.

Displaying all 2^3 valuations for the context $\Gamma = \mathtt{p}_1, \mathtt{p}_2, \mathtt{p}_3$, therefore results in a table that checks for validity of ϕ,

$\mathtt{p}_1$	$\mathtt{p}_2$	$\mathtt{p}_3$	$\mathtt{p}_3$	$\Leftrightarrow$	$\neg$	$\mathtt{p}_1$	$\vee$	$\mathtt{p}_2$	$\wedge$	$\bot$	$\vee$	$\mathtt{p}_1$	$\Rightarrow$	$\mathtt{p}_3$
1	1	1	.	1	...									
1	1	0	.	1		...								
1	0	1	1	1	0	1	0	0	0	0	1	1	1	1
1	0	0	.	1				...						
0	1	1	.	1					...					
0	1	0	.	1						...				
0	0	1	.	1							...			
0	0	0	.	1								...		

In this case, working out the other rows shows that ϕ is indeed valid, thus $\vDash \phi$.

Theorem 2.2.1 (Soundness and Completeness of Propositional Calculus)**.** *Let Φ be any set of formulas and ϕ any formula, then*

$$\Phi \vdash \phi \iff \Phi \vDash \phi.$$

In particular, for any propositional formula ϕ we have

$$\vdash \phi \iff \vDash \phi.$$

Thus derivability and validity coincide.

Proof. Let us sketch the usual proof, for later reference.

(*Soundness*:) First assume $\Phi \vdash \phi$ is provable, meaning there is a finite derivation of $\Phi \vdash \phi$ by the rules of inference. We show by induction on the set of derivations that $\Phi \vDash \phi$, meaning that for any valuation v such that $v \vDash \Phi$ also $v \vDash \phi$. For this, observe that in each individual rule of inference, if $\Psi \vDash \psi$ for all the premisses of the rule, then $\Phi \vDash \phi$ for the conclusion (the set of premisses may change from the premisses to the conclusion if the rule involves a cancellation).

(*Competeness*:) Suppose that $\Phi \nvdash \phi$, then $\Phi, \neg\phi \nvdash \bot$ (using double negation elimination). By Lemma 2.2.2 below, there is a valuation v such that $v \vDash \{\Phi, \neg\phi\}$. Thus in particular $v \vDash \Phi$ and $v \nvDash \phi$, therefore $\Phi \nvDash \phi$. □

The key lemma is this:

Lemma 2.2.2 (Model Existence)**.** *If a set Φ of formulas is* consistent*, in the sense that $\Phi \nvdash \bot$, then it has a* model*, i.e. a valuation v such that $v \vDash \Phi$.*

Proof. Let Φ be any consistent set of formulas. We extend $\Phi \subseteq \Psi$ to one that is *maximally consistent*, meaning Ψ is consistent, and if $\Psi \subseteq \Psi'$ and Ψ' is consistent, then $\Psi = \Psi'$. Enumerate the formulas $\phi_0, \phi_1, \ldots$, and let,

$$\begin{aligned} \Phi_0 &= \Phi, \\ \Phi_{n+1} &= \Phi_n \cup \phi_n \text{ if consistent, else } \Phi_n, \\ \Psi &= \textstyle\bigcup_n \Phi_n. \end{aligned}$$

One can then show that Ψ is indeed maximally consistent, and for every formula ψ, either $\psi \in \Psi$ or $\neg\psi \in \Psi$ and not both (exercise!). Now for each propositional variable $\mathtt{p}$, define $v_\Psi(\mathtt{p}) = 1$ just if $\mathtt{p} \in \Psi$. Finally, one shows that $[\![\phi]\!]^{v_\Psi} = 1$ just if $\phi \in \Psi$, and therefore $v_\Psi \vDash \Psi \supseteq \Phi$. □

Exercise 2.2.3. Show that for any maximally consistent set Ψ of formulas, either $\psi \in \Psi$ or $\neg\psi \in \Psi$ and not both. Conclude from this that for the valuation v_Ψ defined by $v_\Psi(\mathtt{p}) = 1$ just if $\mathtt{p} \in \Psi$, we indeed have $[\![\phi]\!]^{v_\Psi} = 1$ just if $\phi \in \Psi$, as claimed in the proof of the Model Existence Lemma 2.2.2.

2.3 Boolean algebra

There is of course another approach to propositional logic, which also goes back to the 19th century, namely Boolean algebra. It draws on the analogy between the propositional operations $\neg, \vee, \wedge$ and the arithmetical ones $-, +, \times$.

Definition 2.3.1. A *Boolean algebra* is a set B equipped with the operations:

$$\begin{aligned} 0, 1 &: 1 \to B \\ \neg &: B \to B \\ \wedge, \vee &: B \times B \to B \end{aligned}$$

satisfying the following equations, for all $x, y, z \in B$:

$$\begin{aligned} x \vee x &= x & x \wedge x &= x \\ x \vee y &= y \vee x & x \wedge y &= y \wedge x \\ x \vee (y \vee z) &= (x \vee y) \vee z & x \wedge (y \wedge z) &= (x \wedge y) \wedge z \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) & x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ 0 \vee x &= x & 1 \wedge x &= x \\ 1 \vee x &= 1 & 0 \wedge x &= 0 \\ \neg(x \vee y) &= \neg x \wedge \neg y & \neg(x \wedge y) &= \neg x \vee \neg y \\ x \vee \neg x &= 1 & x \wedge \neg x &= 0 \end{aligned}$$

Familiar examples of Boolean algebras are $\mathbb{2} = \{0, 1\}$ with the usual operations on "truth-values", and more generally, any powerset $\mathcal{P}X$, with the set-theoretic operations $A \vee B = A \cup B,\ A \wedge B = A \cap B,\ \neg A = X \setminus A$ (indeed, $\mathbb{2} \cong \mathcal{P}1$ is a special case). This is of course an algebraic theory, like those considered in the previous chapter. The Lawvere algebraic theory $\mathbb{B}$ of Boolean algebras is then, as we know, the opposite of the full subcategory $\mathsf{BA}_{\mathsf{fgf}} \hookrightarrow \mathsf{BA}$ of finitely generated free algebras $B(n)$. We shall consider this aspect later, and in fact we shall see that $\mathbb{B}$ is equivalent to the category of finite powersets $\mathcal{P}[n]$ and arbitrary functions between them.

One can use equational reasoning in Boolean algebra as an alternative to the deductive propositional calculus as follows. For a propositional formula in context $\Gamma \mid \phi$, let us say that ϕ is *equationally provable* if we can prove $\phi = 1$ by the usual equational reasoning (Section B.5), using the laws of Boolean algebras above. More generally, for a set of formulas Φ and a formula ψ let us define the (*ad hoc*) relation of *equational provability*,

$$\Phi \vdash_{\mathsf{eq}} \psi \tag{2.2}$$

to mean that $\psi = 1$ can be proven equationally from (the Boolean equations and) the set of all equations $\phi = 1$, for $\phi \in \Phi$. Since we don't have any laws for the propositional connectives $\Rightarrow$ or $\Leftrightarrow$, let us replace them with their Boolean equivalents, by adding to the equations we are allowed to use the following two:

$$\begin{aligned}\phi \Rightarrow \psi \ &= \ \neg\phi \vee \psi\,,\\ \phi \Leftrightarrow \psi \ &= \ (\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)\,.\end{aligned}$$

Here for example is an equational proof of $(\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)$.

$$\begin{aligned}(\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi) &= (\neg\phi \vee \psi) \vee (\neg\psi \vee \phi)\\ &= \neg\phi \vee (\psi \vee (\neg\psi \vee \phi))\\ &= \neg\phi \vee ((\psi \vee \neg\psi) \vee \phi)\\ &= \neg\phi \vee (1 \vee \phi)\\ &= \neg\phi \vee 1\\ &= 1 \vee \neg\phi\\ &= 1\end{aligned}$$

This shows that

$$\vdash_{\mathsf{eq}} (\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)\,.$$

We may now ask: *How is equational provability $\Phi \vdash_{\mathsf{eq}} \psi$ related to deductive derivability $\Phi \vdash \psi$ and semantic entailment $\Phi \vDash \psi$?*

Exercise 2.3.2. Show by equational reasoning that an equation $\phi = \psi$ is provable from the laws of Boolean algebra if and only if $\vdash_{\mathsf{eq}} (\phi \Leftrightarrow \psi)$.

Exercise 2.3.3. Using equational reasoning, show that every propositional formula ϕ has both a *conjunctive* $\phi^{\wedge}$ and a *disjunctive* $\phi^{\vee}$ *Boolean normal form* such that:

1. The formula $\phi^{\vee}$ is an n-fold disjunction of m-fold conjunctions of *positive* $\mathtt{p_i}$ or *negative* $\neg\mathtt{p_j}$ propositional variables,

   $$\phi^{\vee} \ = \ (\mathsf{q}_{11} \wedge \ldots \wedge \mathsf{q}_{1m_1}) \vee \ldots \vee (\mathsf{q}_{n1} \wedge \ldots \wedge \mathsf{q}_{nm_n})\,, \qquad \mathsf{q}_{ij} \in \{\mathsf{p}_{ij}, \neg\mathsf{p}_{ij}\}\,,$$

   and $\phi^{\wedge}$ is the same, but with the roles of $\vee$ and $\wedge$ reversed.

2. Both

   $$\vdash_{\mathsf{eq}} \phi \Leftrightarrow \phi^{\vee} \qquad \text{and} \qquad \vdash_{\mathsf{eq}} \phi \Leftrightarrow \phi^{\wedge}\,.$$

(*Hint:* Rewrite the formula in terms of just conjunction, disjunction, and negation, and then prove by structural induction on the formula that it has *both* normal forms.)

Exercise 2.3.4. Show that the free Boolean algebra $B(n)$ on n-many generators has 2^{2^n} many elements. *Hint*: Show first that every element $b \in B(n)$ can be written in a unique (disjunctive) normal form (as in the previous exercise):

$$\begin{aligned}&b = b_1 \vee \ldots \vee b_n,\\ &b_i = a_1 \wedge \ldots \wedge a_m, \quad 1 \leq i \leq n,\end{aligned}$$

where each a_j is an element of the finite set $[n] = \{x_1, \ldots, x_n\}$, written in either positive x or negative $\neg x$ form (and not both). Then count these normal forms.

Remark 2.3.5. We can use Exercise 2.3.3 to show that equational provability is equivalent to semantic validity,

$$\vdash_{\mathsf{eq}} \phi \iff \vDash \phi \,.$$

To show this, we first use equational reasoning to put the formula ϕ into conjunctive normal form, and then read off a truth valuation that falsifies it, just if there is one. Indeed, the CNF is valued as 1 just if each conjunct is, and that evidently holds just if each conjunct contains a propositional letter p in both positive p and negative $\neg\mathsf{p}$ form. In that case, the CNF clearly reduces to 1 by an equational calculation. Conversely, if the CNF does not so reduce, it must have a conjunct that does not satisfy the condition just stated; then we can read off from it a valuation that makes all of the (positive and negative) propositional letters in that conjunct 0.

Corollary 2.3.6 (Soundness and completeness of equational inference)**.** *For any set of formulas Φ and formula ψ, we have an equivalence*

$$\Phi \vdash_{\mathsf{eq}} \psi \iff \Phi \vDash \psi \,.$$

Exercise 2.3.7. Prove this, using Remark 2.3.5 for the case where Φ is empty.

Before showing that equational provability $\Phi \vdash_{\mathsf{eq}} \psi$ is also equivalent to deductive derivability $\Phi \vdash \psi$ we shall consider what can be said about Boolean algebra just from the fact that it is a Lawvere algebraic theory, using what we already know about such theories.

2.4 Lawvere duality for Boolean algebras

Let us apply the machinery of algebraic theories from Chapter 1 to the algebraic theory of Boolean algebras and see what we get. The algebraic theory $\mathbb{B}$ of Boolean algebras is a finite product (FP) category with objects $1, B, B^2, ...$, containing a Boolean algebra $\mathsf{U}_{\mathbb{B}}$, with underlying object $|\mathsf{U}_{\mathbb{B}}| = B$. By Theorem 1.1.21, $\mathbb{B}$ has the universal property that finite product preserving (FP) functors from $\mathbb{B}$ into any FP-category $\mathcal{C}$ correspond (pseudo-)naturally to Boolean algebras in $\mathcal{C}$,

$$\mathsf{Hom}_{\mathsf{FP}}(\mathbb{B}, \mathcal{C}) \simeq \mathsf{BA}(\mathcal{C})\,. \tag{2.3}$$

The correspondence is mediated by evaluating an FP functor $F : \mathbb{B} \to \mathcal{C}$ at (the underlying structure of) the Boolean algebra $\mathsf{U}_{\mathbb{B}}$ to get a Boolean algebra $F(\mathsf{U}_{\mathbb{B}})$ in $\mathcal{C}$:

$$\begin{array}{cc} F : \mathbb{B} \longrightarrow \mathcal{C} & \mathsf{FP} \\ \hline\hline F(\mathsf{U}_{\mathbb{B}}) & \mathsf{BA}(\mathcal{C}) \end{array}$$

We call $\mathsf{U}_{\mathbb{B}}$ the *universal Boolean algebra.* Given a Boolean algebra B in $\mathcal{C}$, we write

$$\mathsf{B}^{\sharp} : \mathbb{B} \longrightarrow \mathcal{C}$$

for the associated *classifying functor*. By the equivalence of categories (2.3), we have isos,

$$\mathsf{B}^\sharp(\mathsf{U}_\mathbb{B}) \cong \mathsf{B}, \qquad F(\mathsf{U}_\mathbb{B})^\sharp \cong F\,.$$

And in particular, $\mathsf{U}_\mathbb{B}^\sharp \cong 1_\mathbb{B} : \mathbb{B} \to \mathbb{B}$.

By Lawvere duality, Corollary 1.2.3, we know that $\mathbb{B}^{\mathsf{op}}$ can be identified with a full subcategory $\mathsf{mod}(\mathbb{B})$ of $\mathbb{B}$-models in Set (i.e. Boolean algebras),

$$\mathbb{B}^{\mathsf{op}} = \mathsf{mod}(\mathbb{B}) \hookrightarrow \mathsf{Mod}(\mathbb{B}) = \mathsf{BA}(\mathsf{Set})\,, \tag{2.4}$$

namely, that consisting of the finitely generated free Boolean algebras $F(n)$. In Exercise 2.3.4, we determined $F(n)$ as having the underlying set $PP[n]$ for an n-element set $[n]$, with the Boolean operation of $\vee$ coming from the (outer) powerset, and the $\wedge$ coming from the inner one, with the generators $\{\{x_i\}\}$ for $x_i \in [n]$. Composing (2.4) and (2.3), we have an embedding of $\mathbb{B}^{\mathsf{op}}$ into the functor category,

$$\mathbb{B}^{\mathsf{op}} \hookrightarrow \mathsf{BA}(\mathsf{Set}) \simeq \mathsf{Hom}_{\mathsf{FP}}(\mathbb{B}, \mathsf{Set}) \hookrightarrow \mathsf{Set}^\mathbb{B}\,, \tag{2.5}$$

which, up to isomorphism, is just the (contravariant) Yoneda embedding, taking $B^n \in \mathbb{B}$ to the covariant representable functor $\mathsf{y}^\mathbb{B}(B^n) = \mathsf{Hom}_\mathbb{B}(B^n, -)$ (cf. Theorem 1.2.15).

Now let us consider provability of equations between terms $\phi : B^n \to B$ in the theory $\mathbb{B}$, which are essentially the same as propositional formulas in context $(\mathsf{p}_1, ..., \mathsf{p}_n \mid \phi)$ modulo $\mathbb{B}$-provable equality. The universal Boolean algebra $\mathsf{U}_\mathbb{B}$ is logically generic, in the sense that for any such formulas ϕ, ψ, we have $\mathsf{U}_\mathbb{B} \vDash \phi = \psi$ just if $\mathbb{B} \vdash \phi = \psi$ (Proposition 1.1.17). The latter condition is equational provability from the axioms for Boolean algebras, which was used in the definition of $\vdash_{\mathsf{eq}} \phi$ (cf. 2.2). So we have:

$$\vdash_{\mathsf{eq}} \phi \iff \mathbb{B} \vdash \phi = 1 \iff \mathsf{U}_\mathbb{B} \vDash \phi = 1\,.$$

As we showed in Proposition **??**, the image of the universal model $\mathsf{U}_\mathbb{B}$ under the (FP) *covariant* Yoneda embedding,

$$\mathsf{y}_\mathbb{B} : \mathbb{B} \to \mathsf{Set}^{\mathbb{B}^{\mathsf{op}}}$$

is also a logically generic model, with underlying object $|\mathsf{y}_\mathbb{B}(\mathsf{U}_\mathbb{B})| = \mathsf{Hom}_\mathbb{B}(-, B)$. By Proposition 1.1.28 we can use that fact to restrict attention to Boolean algebras in Set, and in particular, to the finitely generated free ones $F(n)$, when testing for equational provability. Specifically, using the (FP) evaluation functors $\mathsf{eval}_{B^n} : \mathsf{Set}^{\mathbb{B}^{\mathsf{op}}} \to \mathsf{Set}$ for all objects $B^n \in \mathbb{B}$, we can continue the above reasoning as follows:

$$\begin{aligned}
\vdash_{\mathsf{eq}} \phi &\iff \mathbb{B} \vdash \phi = 1 \\
&\iff \mathsf{U}_\mathbb{B} \vDash \phi = 1 \\
&\iff \mathsf{y}_\mathbb{B}(\mathsf{U}_\mathbb{B}) \vDash \phi = 1 \\
&\iff \mathsf{eval}_{B^n}\mathsf{y}_\mathbb{B}(\mathsf{U}_\mathbb{B}) \vDash \phi = 1 \quad \text{for all } B^n \in \mathbb{B} \\
&\iff F(n) \vDash \phi = 1 \quad \text{for all } n.
\end{aligned}$$

The last step holds because the image of $\mathsf{y}_{\mathbb{B}}(\mathsf{U}_{\mathbb{B}})$ under eval_{B^n} is exactly the free Boolean algebra $\mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(\mathsf{U}_{\mathbb{B}}) = F(n)$ (cf. Exercise 1.1.26). Indeed, for the underlying objects we have

$$\mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(\mathsf{U}_{\mathbb{B}}) \cong \mathsf{Hom}_{\mathbb{B}}(B^n, B) \cong \mathsf{Hom}_{\mathsf{BA}^{\mathsf{op}}}(F(n), F(1)) \cong \mathsf{Hom}_{\mathsf{BA}}(F(1), F(n)) \cong |F(n)|\,.$$

Thus to test for equational provability it suffices to check the equations in the free algebras $F(n)$ (which makes sense, since $F(n)$ is usually *defined* in terms of equational provability). We have therefore shown:

Lemma 2.4.1. *A formula in context* $\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi$ *is equationally provable* $\vdash_{\mathsf{eq}} \phi$ *just in case it holds in every finitely generated free Boolean algebra* $F(n)$*,* i.e. $F(n) \vDash \phi = 1$.

Recall that the condition $F(n) \vDash \phi = 1$ means that the equation $\phi = 1$ holds *generally* in $F(n)$, i.e. for any elements $f_1, ..., f_k \in F(n)$, we have $\phi[f_1/\mathsf{p}_1, ..., f_k/\mathsf{p}_k] = 1$, where the expression $\phi[f_1/\mathsf{p}_1, ..., f_k/\mathsf{p}_k]$ denotes the element of $F(n)$ resulting from interpreting the propositional variables p_i as the elements f_i and evaluating the resulting expression using the Boolean operations of $F(n)$. But now observe that the recipe:

> for any elements $f_1, ..., f_k \in F(n)$, let the expression
>
> $$\phi[f_1/\mathsf{p}_1, ..., f_k/\mathsf{p}_k] \tag{2.6}$$
>
> denote the element of $F(n)$ resulting from interpreting the propositional variables p_i as the elements f_i and evaluating the resulting expression using the Boolean operations of $F(n)$

just describes the unique Boolean homomorphism

$$\begin{tikzcd} F(1) \arrow[r, "\overline{\phi}"] & F(k) \arrow[r, "\overline{(f_1, ..., f_k)}"] & F(n)\,, \end{tikzcd}$$

where $\overline{(f_1, ..., f_k)} : F(k) \to F(n)$ is determined by the elements $f_1, ..., f_k \in F(n)$, and $\overline{\phi} : F(1) \to F(k)$ by the corresponding element $(\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi) \in F(k)$. It is therefore equivalent to check the case $k = n$ and $f_i = \mathsf{p}_i$, i.e. the “universal case”

$$(\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi) = 1 \quad \text{in } F(k)\,. \tag{2.7}$$

Finally, then, we have:

Proposition 2.4.2 (Boolean-valued completeness of the equational propositional calculus)**.** *Equational propositional calculus is sound and complete with respect to* boolean-valued models *in* Set*, in the sense that a propositional formula* ϕ *is equationally provable from the laws of Boolean algebra,*

$$\vdash_{\mathsf{eq}} \phi\,,$$

just if it holds generally in any Boolean algebra (in Set*), which we may denote*

$$\vDash_{\mathsf{BA}} \phi\,.$$

Proof. By "holding generally" is meant that it holds for all elements of the Boolean algebra B, in the sense displayed after the Lemma. But, as above, this is equivalent to the condition that for all $b_1, ..., b_k \in \mathsf{B}$, for $\overline{(b_1, ..., b_k)} : F(k) \to \mathsf{B}$ we have $\overline{(b_1, ..., b_k)}(\phi) = 1$ in B, which in turn is clearly equivalent to the previously determined "universal" condition (2.7) that $\phi = 1$ in $F(k)$. □

We leave the analogous statement for equational entailment $\Phi \vdash_{\mathsf{eq}} \phi$ and Boolean-valued entailment $\Phi \vDash_{\mathsf{BA}} \phi$ as an exercise.

Corollary 2.4.3. *Show that a propositional formula $\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi$ is equationally provable $\vdash_{\mathsf{eq}} \phi$, just if it holds in the free Boolean algebra $F(\omega)$ on countably many generators $\omega = \{\mathsf{p}_1, \mathsf{p}_2, ...\}$, with the variables $\mathsf{p}_1, ..., \mathsf{p}_k$ interpreted as the corresponding generators of $F(\omega)$.*

Exercise 2.4.4. Prove this as an easy corollary of Proposition 2.4.2.

Let us summarize what we know so far. By Exercise **??**, we already knew that equational provability in Boolean algebra is equivalent to semantic validity,

$$\vdash_{\mathsf{eq}} \phi \iff \vDash \phi \,.$$

This was based on a certain *decision procedure* for validity in classical propositional logic, originally due to Bernays [?], restated in terms of Boolean algebra. Like the classical proof of the Completeness Theorem 2.2.1,

$$\vdash \phi \iff \vDash \phi \,,$$

we would like to analyze this result, too, in general categorical terms, in order to be able to extend and generalize it to other systems of logic.

Our algebraic approach via Lawvere duality resulted in Proposition 2.4.2, which says that equational provability is equivalent to what we have called *Boolean-valued validity*,

$$\vdash_{\mathsf{eq}} \phi \iff \vDash_{\mathsf{BA}} \phi \iff \mathsf{B} \vDash \phi \quad \text{for all } \mathsf{B}\,. \tag{2.8}$$

This is essentially the Boolean algebra case of our Proposition 1.1.28, the completeness of equational reasoning with respect to algebras in Set, originally proved by Birkhoff.

It still remains to relate equational provability $\vdash_{\mathsf{eq}} \phi$ with deduction $\vdash \phi$, and Boolean-valued validity $\vDash_{\mathsf{BA}} \phi$ with semantic validity $\vDash \phi$, which is just the special case $2 \vDash_{\mathsf{BA}} \phi$. We shall consider deduction $\vdash \phi$ via a different approach in the following section, one that regards Boolean algebras as special finite product categories, rather than as algebras for a special Lawvere algebraic theory.

Exercise 2.4.5. For a formula in context $\mathsf{p}_1, ..., \mathsf{p}_k \mid \vartheta$ and a Boolean algebra B, let the expression $\vartheta[b_1/\mathsf{p}_1, ..., b_k/\mathsf{p}_k]$ denote the element of B resulting from interpreting the propositional variables p_i in the context as the elements b_i of B, and evaluating the resulting expression using the Boolean operations of B. For any *finite* set of propositional formulas

Φ and any formula ψ, let $\Gamma = \mathtt{p}_1, ..., \mathtt{p}_k$ be a context for (the formulas in) $\Phi \cup \{\psi\}$. Finally, recall that $\Phi \vdash_{\mathsf{eq}} \psi$ means that $\psi = 1$ is equationally provable from the set of equations $\{\phi = 1 \mid \phi \in \Phi\}$. Show that $\Phi \vdash_{\mathsf{eq}} \psi$ just if for all finitely generated free Boolean algebras $F(n)$, the following condition holds:

> For any elements $f_1, ..., f_k \in F(n)$, if $\phi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] = 1$ for all $\phi \in \Phi$, then $\psi[f_1/\mathtt{p}_1, ..., f_k/\mathtt{p}_k] = 1$.

Is it sufficient to just take $F(k)$ and its generators $\mathtt{p}_1, ..., \mathtt{p}_k$ as the $f_1, ..., f_k$? Is it equivalent to take all Boolean algebras B, rather than the finitely generated free ones $F(n)$? Determine a condition that is equivalent to $\Phi \vdash_{\mathsf{eq}} \psi$ for not necessarily finite sets Φ.

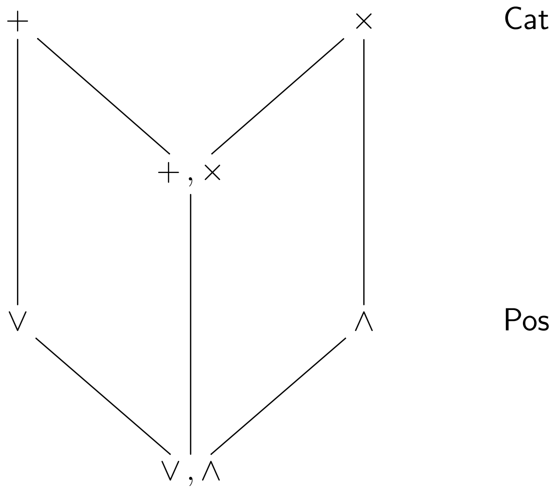
Exercise 2.4.6. A Boolean algebra can be partially ordered by defining $x \leq y$ as

$$x \leq y \iff x \vee y = y \qquad \text{or equivalently} \qquad x \leq y \iff x \wedge y = x\,.$$

Thus a Boolean algebra is a (poset) category. Show that as a category, a Boolean algebra has all finite limits and colimits and is cartesian closed, with $x \Rightarrow y := \neg x \vee y$ as the exponential of x and y. Moreover, a finitely complete and cocomplete cartesian closed poset is a Boolean algebra just if it satisfies $x = (x \Rightarrow 0) \Rightarrow 0$. Finally, show that homomorphisms of Boolean algebras $f : B \to B'$ are exactly the same thing as functors (i.e. monotone maps) that preserve all finite limits and colimits.

2.5 Functorial semantics for propositional logic

Considering the algebraic theory of Boolean algebras suggested the idea of a Boolean valuation of propositional logic, generalizing the truth valuations of section 2.2. This can be seen as applying the framework of functorial semantics to a different system of logic than that of equational theories, represented as finite product categories, namely that represented categorically by *poset* categories with finite products $\wedge$ *and coproducts* $\vee$ (each of these cases could, of course, also be considered separately, relating $\wedge$-semilattices and categories with finite products $\times$, and $\vee$-semilattices with categories with coproducts $+$, respectively). Thus we are moving from the top right corner to the bottom center position in the following Hasse diagram of structured categories:



In Chapter **??** we shall see how first-order logic results categorically from these two cases by "indexing the lower one over the upper one", in a certain sense, and in Chapters **??** and **??** we shall consider simple and dependent type theory as "categorified" versions of propositional and first-order logic. It is for this reason (rather than a dogmatic commitment to categorical methods!) that we next continue our reformulation of the basic results of classical propositional logic in functorial terms.

Exercise 2.5.1. Review the results of Chapter 1 on (Lawvere) algebraic theories in the case of posets. First, show that a *posetal* Lawvere algebraic theory is always trivial (*why?*), but that a (general) posetal algebraic theory is a $\wedge$-semilattice. What are the Set-valued models of such a theory? What do the duality theories of Chapter 1 mean in this setting?

Definition 2.5.2. A *propositional theory* $\mathbb{T}$ consists of a set $V_{\mathbb{T}}$ of propositional variables, called the *basic* or *atomic propositions*, and a set $A_{\mathbb{T}}$ of propositional formulas (over $V_{\mathbb{T}}$), called the *axioms*. The *(deductive) consequences* $\Phi \vdash_{\mathbb{T}} \phi$ are those judgements that are derivable by natural deduction (as in Section 2.1), from the axioms $A_{\mathbb{T}}$, where we define $\Phi \vdash_{\mathbb{T}} \phi$ to mean $\Phi \cup A_{\mathbb{T}} \vdash \phi$ for (sets of) formulas Φ, ϕ over $V_{\mathbb{T}}$.

Definition 2.5.3. Let $\mathbb{T} = (V_{\mathbb{T}}, A_{\mathbb{T}})$ be a propositional theory and $\mathcal{B}$ a Boolean algebra. A *model* of $\mathbb{T}$ in $\mathcal{B}$, also called a *Boolean valuation of* $\mathbb{T}$ is an *interpretation function* $v : V_{\mathbb{T}} \to |\mathcal{B}|$ such that, for every $\alpha \in A_{\mathbb{T}}$, we have $[\![\alpha]\!]^v = 1_{\mathcal{B}}$ in $\mathcal{B}$, where the extension

$[\![-]\!]^v$ of v from $V_{\mathbb{T}}$ to all formulas (over $V_{\mathbb{T}}$) is defined in the expected way, namely:

$$\begin{aligned}
[\![\mathsf{p}]\!]^v &= v(\mathsf{p}), \quad \mathsf{p} \in V_{\mathbb{T}} \\
[\![\top]\!]^v &= 1_{\mathcal{B}} \\
[\![\bot]\!]^v &= 0_{\mathcal{B}} \\
[\![\neg\phi]\!]^v &= \neg_{\mathcal{B}}[\![\phi]\!]^v \\
[\![\phi \wedge \psi]\!]^v &= [\![\phi]\!]^v \wedge_{\mathcal{B}} [\![\psi]\!]^v \\
[\![\phi \vee \psi]\!]^v &= [\![\phi]\!]^v \vee_{\mathcal{B}} [\![\psi]\!]^v \\
[\![\phi \Rightarrow \psi]\!]^v &= \neg_{\mathcal{B}}[\![\phi]\!]^v \vee_{\mathcal{B}} [\![\psi]\!]^v
\end{aligned}$$

Finally, let $\mathsf{Mod}(\mathbb{T}, \mathcal{B})$ be the set of all $\mathbb{T}$-models in $\mathcal{B}$. Given a Boolean homomorphism $f : \mathcal{B} \to \mathcal{B}'$, there is an induced mapping $\mathsf{Mod}(\mathbb{T}, f) : \mathsf{Mod}(\mathbb{T}, \mathcal{B}) \to \mathsf{Mod}(\mathbb{T}, \mathcal{B}')$, determined by setting $\mathsf{Mod}(\mathbb{T}, f)(v) = f \circ v$, which is clearly functorial.

Theorem 2.5.4. *The functor* $\mathsf{Mod}(\mathbb{T}) : \mathsf{BA} \to \mathsf{Set}$ *is representable, with representing Boolean algebra* $\mathcal{B}_{\mathbb{T}}$*, the* classifying Boolean algebra *of* $\mathbb{T}$*. Thus there is a natural iso,*

$$\mathsf{Hom}_{\mathsf{BA}}(\mathcal{B}_{\mathbb{T}}, \mathcal{B}) \;\cong\; \mathsf{Mod}(\mathbb{T}, \mathcal{B})\,. \tag{2.9}$$

Proof. We construct $\mathcal{B}_{\mathbb{T}}$ from the "syntax of $\mathbb{T}$" in two steps:

Step 1: Suppose first that $A_{\mathbb{T}}$ is empty, so $\mathbb{T}$ is just a set V of propositional variables. Then define the classifying Boolean algebra $\mathcal{B}[V]$ by

$$\mathcal{B}[V] \;=\; \{\phi \mid \phi \text{ is a formula in context } V\}/\sim$$

where the equivalence relation $\sim$ is *(deductively) provable bi-implication*,

$$\phi \sim \psi \iff \vdash \phi \Leftrightarrow \psi.$$

The operations are (well-)defined on equivalence classes by setting,

$$[\phi] \wedge [\psi] \;=\; [\phi \wedge \psi],$$

and so on. (The reader who has not seen this construction before should fill in the details!)

Step 2: In the general case $\mathbb{T} = (V_{\mathbb{T}}, A_{\mathbb{T}})$, let

$$\mathcal{B}_{\mathbb{T}} = \mathcal{B}[V_{\mathbb{T}}]/\sim_{\mathbb{T}}\,,$$

where the equivalence relation $\sim_{\mathbb{T}}$ is now $A_{\mathbb{T}}$*-provable bi-implication*,

$$\phi \sim_{\mathbb{T}} \psi \iff A_{\mathbb{T}} \vdash \phi \Leftrightarrow \psi.$$

The operations are defined as before, but now on equivalence classes $[\phi]$ modulo $A_{\mathbb{T}}$.

Observe that the construction of $\mathcal{B}_{\mathbb{T}}$ is a variation on that of the *syntactic category* construction $\mathcal{C}_{\mathbb{T}} = \mathsf{Syn}(\mathbb{T})$ of the classifying category of an algebraic theory $\mathbb{T}$, in the sense of the previous chapter. Indeed, the statement of the theorem (2.9) is exactly the universal property of $\mathcal{B}_{\mathbb{T}}$ as the classifying category of $\mathbb{T}$-models. (Since $\mathsf{Mod}(\mathbb{T}, \mathcal{B})$ is now a *set* rather than a category, we can classify it up to *isomorphism* rather than equivalence of categories.) The proof of this fact is a variation on the proof of the corresponding theorem 1.1.21 from Chapter 1. Further details are given in the following Remark 2.5.6 for the interested reader. □

Remark 2.5.5. The Lindenbaum-Tarski algebra of a propositional theory is usually defined in semantic terms using (truth) valuations. Our definition of $\mathcal{B}_{\mathbb{T}}$ in terms of *provability* is more useful in the present setting, as it parallels that of the syntactic category $\mathcal{C}_{\mathbb{T}}$ of an algebraic theory, and will allow us to prove Theorem 2.2.1 by analogy to Theorem **??** for algebraic theories.

Remark 2.5.6 (Adjoint Rules for Propositional Calculus)**.** For the construction of the classifying algebra $\mathcal{B}_{\mathbb{T}}$, it is convenient to reformulate the rules of inference for the propositional calculus in the following equivalent *adjoint form*: Contexts Γ may be omitted, since the rules leave them unchanged (there is no variable binding). We may also omit assumptions that remain unchanged. Thus e.g. the *hypothesis* rule may be written in any of the following equivalent ways.

$$\frac{}{\Gamma \mid \phi_1, \ldots, \phi_m \vdash \phi_i} \qquad \frac{}{\phi_1, \ldots, \phi_m \vdash \phi_i} \qquad \frac{}{\phi \vdash \phi}$$

The structural rules can then be stated as follows:

$$\frac{}{\phi \vdash \phi} \qquad \frac{\phi \vdash \psi \qquad \psi \vdash \vartheta}{\phi \vdash \vartheta}$$

$$\frac{\phi \vdash \vartheta}{\psi, \phi \vdash \vartheta} \qquad \frac{\phi, \phi \vdash \vartheta}{\phi \vdash \vartheta} \qquad \frac{\phi, \psi \vdash \vartheta}{\psi, \phi \vdash \vartheta}$$

The rules for the propositional connectives can be given in the following adjoint form, where the double line indicates a two-way rule (with the obvious two instances when there are two conclusions, in going from bottom to top).

$$\frac{}{\phi \vdash \top} \qquad \frac{}{\bot \vdash \phi}$$

$$\frac{\vartheta \vdash \phi \quad \vartheta \vdash \psi}{\vartheta \vdash \phi \wedge \psi} \qquad \frac{\phi \vdash \vartheta \quad \psi \vdash \vartheta}{\phi \vee \psi \vdash \vartheta} \qquad \frac{\vartheta, \phi \vdash \psi}{\vartheta \vdash \phi \Rightarrow \psi}$$

For the purpose of deduction, negation $\neg\phi$ is again treated as defined by $\phi \Rightarrow \bot$ and bi-implication $\phi \Leftrightarrow \psi$ by $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$.

For *classical* logic we also include the rule of *double negation*:

$$\frac{}{\neg\neg\phi \vdash \phi} \tag{2.10}$$

It is now obvious that the set of formulas is preordered by $\phi \vdash \psi$, and that the poset reflection agrees with the deducibility equivalence relation,

$$\phi \dashv\vdash \psi \iff \phi \sim \psi \,.$$

Moreover, $\mathcal{B}_{\mathbb{T}}$ clearly has all finite limits $\top, \wedge$ and colimits $\bot, \vee$, is cartesian closed $\wedge \dashv \Rightarrow$, and is therefore a *Heyting algebra* (see Section **??** below). The rule of double negation then makes it a Boolean algebra.

The proof of the universal property of $\mathcal{B}_{\mathbb{T}}$ is essentially the same as that for $\mathcal{C}_{\mathbb{T}}$.

Exercise 2.5.7. Fill in the details of the proof that $\mathcal{B}_{\mathbb{T}}$ is a well-defined Boolean algebra, with the universal property stated in (2.9). (*Hint:* The well-definedness of the operations $[\phi] \wedge [\psi]$, etc., just requires a few deductions, but the well-definedness of the Boolean homomorphism $v^\sharp : \mathcal{B}_{\mathbb{T}} \to \mathcal{B}$ classifying a model $v : V_{\mathbb{T}} \to |\mathcal{B}|$ requires the *soundness* of deduction with respect to Boolean-valued semantics. Just state this precisely and sketch a proof of it.)

Just as for the case of algebraic theories and FP categories, we now have the following corollary of the classifying theorem 2.5.4, which again follows from the fact that the classifying Boolean algebra $\mathcal{B}_{\mathbb{T}}$ is *logically generic*, in virtue of its syntactic construction.

Corollary 2.5.8. *For any formula ϕ, derivability from the axioms $A_{\mathbb{T}} \vdash \phi$ is equivalent to validity under all Boolean-valued models of $\mathbb{T}$,*

$$A_{\mathbb{T}} \vdash \phi \iff A_{\mathbb{T}} \vDash_{\mathsf{BA}} \phi \,.$$

where, recall, $A_{\mathbb{T}} \vDash_{\mathsf{BA}} \phi$ means that for all Boolean algebras $\mathcal{B}$ and valuations $v : V_{\mathbb{T}} \to |\mathcal{B}|$ such that $[\![\alpha]\!]^v = 1$ in $\mathcal{B}$ for all $\alpha \in A_{\mathbb{T}}$, we also have $[\![\phi]\!]^v = 1$ in $\mathcal{B}$.

Proof. We have

$$A_{\mathbb{T}} \vdash \phi \iff \mathcal{B}_{\mathbb{T}} \vDash_{\mathsf{BA}} \phi \,,$$

essentially by definition, where on the righthand side it suffices to check the canonical model $u : V_{\mathbb{T}} \to |\mathcal{B}_{\mathbb{T}}|$ associated to the identity $\mathcal{B}_{\mathbb{T}} \to \mathcal{B}_{\mathbb{T}}$. But if $u \vDash_{\mathsf{BA}} \phi$, then also $v \vDash_{\mathsf{BA}} \phi$ for any $v : V_{\mathbb{T}} \to |\mathcal{B}|$, since $v = v^\sharp u$, and the homomorphism $v^\sharp : \mathcal{B}_{\mathbb{T}} \to \mathcal{B}$ preserves models. Thus $\mathcal{B}_{\mathbb{T}} \vDash_{\mathsf{BA}} \phi \Rightarrow A_{\mathbb{T}} \vDash_{\mathsf{BA}} \phi$. The converse is immediate. □

Now note that the recipe displayed at (2.6) for a Boolean valuation in $F(n)$ of a formula in context $\mathsf{p}_1, ..., \mathsf{p}_k \mid \phi$ is exactly the (canonical) *model* in $F(n)$, with underlying valuation $\eta : \{\mathsf{p}_1, ..., \mathsf{p}_k\} \to |F(n)|$, of the theory $\mathbb{T} = \{\mathsf{p}_1, ..., \mathsf{p}_k\}$. So

$$F(n) \vDash_{\mathsf{BA}} \phi \iff [\![\phi]\!] = 1 \text{ in } F(n) \,.$$

Inspecting the universal property (2.9) of $\mathcal{B}_{\mathbb{T}}$ for the case $\mathbb{T} = \{\mathsf{p}_1, ..., \mathsf{p}_n\}$, we also obtain:

Corollary 2.5.9. *The classifying Boolean algebra for the theory $\{\mathsf{p}_1, ..., \mathsf{p}_n\}$ is the finitely generated, free Boolean algebra,*

$$\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \;\cong\; F(n)\,.$$

And generally, $\mathcal{B}[V]$ is the free Boolean algebra on the set V, for any set V.

Indeed, for any valuation (= arbitrary function) $v : \{\mathsf{p}_1, ..., \mathsf{p}_n\} \to |\mathcal{B}|$ we have a unique extension $[\![-]\!]^v : \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \to \mathcal{B}$, which upon inspection of Definition 2.5.3 we recognize as exactly a Boolean homomorphism.

$$\begin{tikzcd}
\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \arrow[r, "{[\![-]\!]^v}"] & \mathcal{B} \\
\{\mathsf{p}_1, ..., \mathsf{p}_n\} \arrow[u, "\eta"] \arrow[ur, "v"'] &
\end{tikzcd}$$

The isomorphism $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \cong F(n)$ of Corollary 2.5.8 expresses the fact that the relations of derivability by natural deduction $\Phi \vdash \phi$ and equational provability $\Phi \vdash_{\mathsf{eq}} \phi$ agree,

$$\Phi \vdash \phi \quad\Longleftrightarrow\quad \Phi \vdash_{\mathsf{eq}} \phi\,, \tag{2.11}$$

answering one of the two questions from the end of Section 2.4.

Toward answering the other question of the relation between Boolean-valued validity $\Phi \vDash_{\mathsf{BA}} \phi$ and truth-valued validity $\Phi \vDash \phi$, consider the *finitely presented* Boolean algebras, which can be described as those of the form

$$\mathcal{B}_{\mathbb{T}} = \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha\,,$$

for a finite theory $\mathbb{T} = (\mathsf{p}_1, ..., \mathsf{p}_n; \alpha_1, ..., \alpha_m)$, where the slice category of a Boolean algebra $\mathcal{B}$ over an element $\beta \in \mathcal{B}$ is the *downset* (or *principal ideal*)

$$\mathcal{B}/\beta = \downarrow(\beta) = \{b \in \mathcal{B} \mid b \leq \beta\}\,.$$

To see this, given $\mathbb{T} = (V_{\mathbb{T}}, A_{\mathbb{T}})$, if $A_{\mathbb{T}}$ is finite, then let

$$\alpha_{\mathbb{T}} \;:=\; \bigwedge_{\alpha \in A_{\mathbb{T}}} \alpha\,,$$

so we clearly have

$$\mathcal{B}_{\mathbb{T}} = \mathcal{B}[V_{\mathbb{T}}]/\alpha_{\mathbb{T}}\,.$$

If $V_{\mathbb{T}} = \{\mathsf{p}_1, ..., \mathsf{p}_n\}$ is also finite, then we have

$$\mathcal{B}_{\mathbb{T}} \cong \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha_{\mathbb{T}}\,.$$

Using this, it is now easy to show that the finitely presented objects in the category of Boolean algebras are exactly those of the form $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha_{\mathbb{T}}$, using the fact that a (Boolean) algebra A is finitely presented if and only if it has a presentation (by n-many generators and m-many equations) as a coequalizer of finitely generated free algebras,

$$\begin{tikzcd}
F(m) \arrow[r, shift left] \arrow[r, shift right] & F(n) \arrow[r] & A\,.
\end{tikzcd} \tag{2.12}$$

Exercise 2.5.10. Show that the classifying Boolean algebras $\mathcal{B}_{\mathbb{T}}$, for *finite sets* $V_{\mathbb{T}}$ of variables and $A_{\mathbb{T}}$ of formulas, are exactly the *finitely presented* ones in the sense stated in (2.12) (*Hint:* Recall that for elements ϕ, ψ in any Boolean algebra, $\phi = \psi$ iff $(\phi \Leftrightarrow \psi) = 1$). In general algebraic categories $\mathcal{A}$ such coequalizers of finitely generated free algebras are exactly those for which the representable functor $\mathsf{Hom}(A, -) : \mathcal{A} \to \mathsf{Set}$ preserves all filtered colimits. Show that the finitely presented Boolean algebras in the sense of (2.12) do indeed have this property. (You need not show the converse, but think about it!)

The following is a special case of the universal property of the slice category

$$X^* : \mathbb{C} \to \mathbb{C}/_X\,,$$

for any $\mathbb{C}$ with finite limits. The reader not already familiar with this fact should definitely do the exercise!

Exercise 2.5.11. For any Boolean algebra $\mathcal{B}$ and any $\beta \in \mathcal{B}$, consider the map

$$\beta^* : \mathcal{B} \to \mathcal{B}/\beta\,,$$

with $\beta^*(x) = \beta \wedge x$.

(i) Show that $\mathcal{B}/\beta \cong \downarrow(\beta)$ is a Boolean algebra, and that β^* is a Boolean homomorphism with $\beta^*(\beta) = 1 \in \mathcal{B}/\beta$.

(ii) If $h : \mathcal{B} \to \mathcal{B}'$ is any homomorphism, then $h(\beta) = 1 \in \mathcal{B}'$ if and only if there is a factorization

$$\begin{tikzcd}
\mathcal{B} \arrow[r, "h"] \arrow[d, "\beta^*"'] & \mathcal{B}' \\
\mathcal{B}/\beta \arrow[ur, "\overline{h}"'] &
\end{tikzcd} \tag{2.13}$$

of h through β^*, and then $\overline{h}$ is unique with $\overline{h} \circ \beta^* = h$.

(iii) Show that if $\mathcal{B}_{\mathbb{T}} = \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha$ classifies (models of) the theory $\mathbb{T} = (\mathsf{p}_1 ... \mathsf{p}_n, \alpha)$ and $\mathsf{p}_1, ..., \mathsf{p}_n \mid \beta$, then $\mathcal{B}_{\mathbb{T}}/\beta$ classifies models of the extended theory $\mathbb{T}' = (\mathsf{p}_1 ... \mathsf{p}_n, \alpha, \beta)$.

Lemma 2.5.12. *Let $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha$ be a* finitely presented *Boolean algebra which is* non-trivial*, in the sense that $0 \neq 1$. Then there is a Boolean homomorphism*

$$h : \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha \to \mathbb{2}\,.$$

Proof. By Exercise 2.5.10, we can assume that $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha = \mathcal{B}_{\mathbb{T}}$ classifies (models of) the theory $\mathbb{T} = (\mathsf{p}_1, ..., \mathsf{p}_n; \alpha)$. By the assumption that $0 \neq 1$ in $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\alpha$, we must have $\alpha \neq 0$ in the free Boolean algebra $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]$ (*why?*). It then suffices to give a valuation $v : \{\mathtt{p}_1, ..., \mathtt{p}_n\} \to 2$ such that $[\![\alpha]\!]^v = 1$, for then (by Exercise 2.5.11) we will have a factorization,

$$\begin{tikzcd}
\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \arrow[r, "m_v"] \arrow[d, "\alpha^*"'] & \mathbb{2} \\
\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\alpha \arrow[ur, "\overline{m_v}"'] &
\end{tikzcd} \tag{2.14}$$

where $m_v = [\![-]\!]^v$ is the "model" associated to the valuation $v : \{\mathtt{p}_1, ..., \mathtt{p}_n\} \to 2$, and $\alpha^* = \alpha \wedge - : \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \longrightarrow \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\alpha$ is the canonical Boolean projection to the "quotient" Boolean algebra given by the slice category, and $\overline{m_v}$ is the extension of m_v along α^* resulting from the universal property of slicing a category with finite products.

Informally, α has a truth table with 2^n rows, corresponding to the valuations $v : \{\mathtt{p}_1, ..., \mathtt{p}_n\} \to 2$, and we know that the main column for α is not all 0's, so we can find a row in which it is 1 and read off the corresponding valuation. More formally, as in Remark 2.3.5, we can put α into a disjunctive normal form $\alpha = \alpha_1 \vee ... \vee \alpha_k$ and one of the disjuncts α_i must then also be non-zero. Since $\alpha_i = q_1 \wedge ... \wedge q_m$ with each q_j either positive $\mathtt{p}$ or negative $\neg\mathtt{p}$, if both $\mathtt{p}$ and $\neg\mathtt{p}$ occur, then $\alpha_i = 0$, so the $\mathtt{p}$ in each q_j must occur only once in α_i. We can then define v accordingly, with $v(\mathtt{p}) = 1$ iff $\mathtt{p}$ occurs positively in α_i, and we will have $[\![\alpha_i]\!]^v = 1$. This valuation $v : \{\mathtt{p}_1, ..., \mathtt{p}_n\} \to 2$ then determines a Boolean homomorphism $[\![-]\!]^v : \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \to \mathbb{2}$ with $[\![\alpha]\!]^v = 1$, as required for a homomorphism

$$\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\alpha \longrightarrow \mathbb{2}\,.$$

□

Proposition 2.5.13. *For any formula ϕ, Boolean-valued validity and truth-valued validity are equivalent,*

$$\vDash_{\mathsf{BA}} \phi \iff \vDash \phi\,. \tag{2.15}$$

Proof. Since $\vDash_{\mathsf{BA}} \phi$ means that $\mathcal{B} \vDash_{\mathsf{BA}} \phi$ for all Boolean algebras $\mathcal{B}$, and $\vDash \phi$ means the same for valuations in $\mathbb{2}$, the implication from left to right is trivial. For the converse, let $(\mathtt{p}_1, ..., \mathtt{p}_n \,|\, \phi)$, and consider $\phi \in \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]$.

Suppose $h(\phi) = 1$ for all homomorphisms $h : \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \to \mathbb{2}$. Then $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\neg\phi$ can have no homomorphism $\overline{h} : \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\neg\phi \to \mathbb{2}$, for otherwise $\overline{h}(1) = 1$ would give $h(\neg\phi) = 1$, and so $h(\phi) = 0$. Therefore, by Lemma 2.5.12, $0 = 1$ in $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]/\neg\phi$. So in $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]$ we have $0 = \neg\phi \wedge 1 = \neg\phi$ whence $1 = \neg 0 = \neg\neg\phi = \phi$, and so $h(\phi) = 1 \in \mathcal{B}$ for all $h : \mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \to \mathcal{B}$. □

Exercise 2.5.14. Extend Proposition 2.5.15 to entailment, for any finite set Φ,

$$\Phi \vDash_{\mathsf{BA}} \phi \iff \Phi \vDash \phi\,.$$

Combining this last result (2.15) with the previous one (2.11) and (2.8) from the last section, we arrive finally at our desired reconstruction of the classical completeness theorem:

Proposition 2.5.15. *For any formula ϕ, provability by deduction and truth-valued validity are equivalent,*

$$\vdash \phi \iff \vDash \phi\,. \tag{2.16}$$

And the same holds relative to a set Φ of assumptions.

Let us unwind the foregoing "reproof" into a direct argument, from the present point of view: A formula ϕ in context $\mathsf{p}_1, ..., \mathsf{p}_n \mid \phi$ determines an element in the free Boolean algebra $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]$. If $\vdash \phi$ then $\phi = 1$ in $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]$, so clearly $h(\phi) = 1$ for every $h : \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \to \mathbb{2}$, which means exactly $\vDash \phi$. Conversely, if $\vDash \phi$ then $h(\phi) = 1$ for every $h : \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n] \to \mathbb{2}$, so $\neg\phi$ can have no model in $\mathbb{2}$. Thus $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]/\neg\phi$ must be degenerate, with $0 = 1$. So in $\mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]$ we have $[\bot] = [\neg\phi]$, and therefore $\neg\phi \vdash \bot$, so $\vdash \neg\neg\phi$, so $\vdash \phi$.

The main fact used here is that the finitely generated, free Boolean algebras $\mathcal{B}(n) = \mathcal{B}[\mathsf{p}_1, ..., \mathsf{p}_n]$ have enough Boolean homomorphisms $h : \mathcal{B}(n) \to \mathbb{2}$ to separate any non-zero element $\phi \neq 0$ from 0, in the sense that if $h(\phi) = 0$ for all such h then $\phi = 0$. In other words, the canonical homomorphism

$$\mathcal{B}(n) \longrightarrow \Pi_{h\in\mathsf{BA}(\mathcal{B}(n),\mathbb{2})}\mathbb{2}\,, \tag{2.17}$$

is injective. This is analogous to the proof of completeness of equational deduction for an algebraic theory $\mathbb{T}$, which used an embedding of the syntactic category $\mathcal{C}_\mathbb{T}$ into a power of Set (rather than $\mathbb{2}$),

$$\mathcal{C}_\mathbb{T} \hookrightarrow \mathsf{Set}^{\mathsf{mod}(\mathbb{T})}$$

for a "sufficient set" of models $\mathsf{mod}(\mathbb{T}) \subset \mathsf{FP}(\mathcal{C}_\mathbb{T}, \mathsf{Set})$, namely all those of the form $\mathcal{C}_\mathbb{T}(-, U^n) : {\mathcal{C}_\mathbb{T}}^{\mathsf{op}} = \mathsf{mod}(\mathbb{T}) \to \mathsf{Set}$. For Boolean algebras, the embedding (2.17) will be a step toward the Stone Representation Theorem.

2.6 Stone representation

Regarding a Boolean algebra $\mathcal{B}$ as a category with finite products, consider its Yoneda embedding $\mathsf{y} : \mathcal{B} \hookrightarrow \mathsf{Set}^{\mathcal{B}^{\mathsf{op}}}$. Since the hom-set $\mathcal{B}(x, y)$ is always $\mathbb{2}$-valued, we have a factorization,

$$\mathsf{y} : \mathcal{B} \hookrightarrow \mathbb{2}^{\mathcal{B}^{\mathsf{op}}} \hookrightarrow \mathsf{Set}^{\mathcal{B}^{\mathsf{op}}} \tag{2.18}$$

in which each factor still preserves the finite products (note that the products in $\mathbb{2}$ are preserved by the inclusion $\mathbb{2} \hookrightarrow \mathsf{Set}$, and the products in the functor categories $\mathbb{2}^{\mathcal{B}^{\mathsf{op}}}$ and $\mathsf{Set}^{\mathcal{B}^{\mathsf{op}}}$ are taken pointwise). Indeed, this is an instance of a general fact. In the category $\mathsf{Cat}_\times$ of finite product categories (and $\times$-preserving functors), the inclusion of the full subcategory of posets with $\wedge$ (the $\wedge$-semilattices) has a *right adjoint* R, in addition to the left adjoint L of poset reflection.

$$\begin{tikzcd}
\mathsf{Cat}_\times \arrow[d, bend right=60, "L"'] \arrow[d, bend left=60, "R"] \\
\mathsf{Pos}_\wedge \arrow[u, hook, "i"']
\end{tikzcd}$$

For a finite product category $\mathbb{C}$, the poset $R\mathbb{C}$ is the subcategory $\mathsf{Sub}(1) \hookrightarrow \mathbb{C}$ of subobjects of the terminal object 1 (equivalently, the category of monos $m : M \rightarrowtail 1$). The reason for this is that a $\times$-preserving functor $f : A \to \mathbb{C}$ from a poset A with meets takes every object $a \in A$ to a mono $f(a) \rightarrowtail 1$ in $\mathbb{C}$, since $a = a \wedge a$ implies the following is a product diagram in A.

$$\begin{tikzcd}
a \arrow[r] & 1 \\
a \arrow[u] \arrow[r] & a \arrow[u]
\end{tikzcd}$$

Exercise 2.6.1. Prove this, and use it to verify that $R = \mathsf{Sub}(1)$ is indeed a right adjoint to the inclusion of $\wedge$-semilattices into finite-product categories.

Now the functor category $\mathbb{2}^{\mathcal{B}^{\mathsf{op}}} = \mathsf{Pos}(\mathcal{B}^{\mathsf{op}}, \mathbb{2})$ occurring in (2.18), consists of all *contravariant*, monotone maps $\mathcal{B}^{\mathsf{op}} \to \mathbb{2}$ (which indeed is $\mathsf{Sub}(1) \hookrightarrow \mathsf{Set}^{\mathcal{B}^{\mathsf{op}}}$), and is easily seen to be isomorphic to the poset $\mathsf{Down}(\mathcal{B})$ of all downsets (or "*sieves*") in $\mathcal{B}$: subsets $S \subseteq \mathcal{B}$ that are downward closed, $x \leq y \in S \Rightarrow x \in S$, ordered by subset inclusion $S \subseteq T$. Explicitly, the isomorphism

$$\mathsf{Pos}(\mathcal{B}^{\mathsf{op}}, \mathbb{2}) \;\cong\; \mathsf{Down}(\mathcal{B}) \tag{2.19}$$

is given by taking $f : \mathcal{B}^{\mathsf{op}} \to \mathbb{2}$ to $f^{-1}(1)$ and $S \subseteq \mathcal{B}$ to the function $f_S : \mathcal{B}^{\mathsf{op}} \to \mathbb{2}$ with $f_S(b) = 1 \Leftrightarrow b \in S$. Under this isomorphism, the Yoneda embedding takes an element $b \in \mathcal{B}$ *covariantly* to the *principal downset* $\downarrow b \subseteq \mathcal{B}$ of all $x \leq b$.

Exercise 2.6.2. Show that (2.19) is indeed an isomorphism of posets, and that it sends the Yoneda embedding to the principal sieve mapping, as claimed.

For algebraic theories $\mathbb{A}$, we used the Yoneda embedding to give a completeness theorem for equational logic with respect to Set-valued models, by composing the (faithful) functor $\mathsf{y} : \mathbb{A} \hookrightarrow \mathsf{Set}^{\mathbb{A}^{\mathsf{op}}}$ with the (jointly faithful) evaluation functors $\mathsf{eval}_A : \mathsf{Set}^{\mathbb{A}^{\mathsf{op}}} \to \mathsf{Set}$, for all objects $A \in \mathbb{A}$. This amounts to considering all *covariant* representables $\mathsf{eval}_A \circ \mathsf{y} = \mathbb{A}(A, -) : \mathbb{A} \to \mathsf{Set}$, and observing that these are then (both $\times$-preserving and) jointly faithful.

We can do exactly the same thing for a Boolean algebra $\mathcal{B}$ (which is, after all, a finite product category) to get a jointly faithful family of $\times$-preserving, monotone maps $\mathcal{B}(b, -) : \mathcal{B} \to \mathbb{2}$, i.e. $\wedge$-semilattice homomorphisms. By taking the preimages of $1 \in \mathbb{2}$, such homomorphisms correspond to *filters* in $\mathcal{B}$: (non-empty) "upsets" that are also closed under $\wedge$.

$$\mathsf{Pos}_{\wedge}(\mathcal{B}, \mathbb{2}) \;\cong\; \mathsf{Filters}(\mathcal{B}) \tag{2.20}$$

The representables $\mathcal{B}(b, -)$ now correspond to the *principal filters* $\uparrow b \subseteq \mathcal{B}$.

The *problem* with using this approach for a completeness theorem for *propositional* logic, however, is that such $\wedge$-homomorphisms $\mathcal{B} \to \mathbb{2}$ are not *models*, because they need not preserve the joins $\phi \vee \psi$ (nor the complements $\neg\phi$).

Lemma 2.6.3. *Let $\mathcal{B}, \mathcal{B}'$ be Boolean algebras and $f : \mathcal{B} \to \mathcal{B}'$ a distributive lattice homomorphism. Then f preserves negation, and so is Boolean. The category* BA *of Boolean algebras is thus a* full *subcategory of the category* DLat *of distributive lattices.*

Proof. The complement $\neg b$ is the unique element of $\mathcal{B}$ such that both $b \vee \neg b = 1$ and $b \wedge \neg b = 0$. □

This suggests representing a Boolean algebra $\mathcal{B}$, not by its filters, but by its *prime* filters, which correspond bijectively to distributive lattice homomorphisms $\mathcal{B} \to \mathbb{2}$.

Definition 2.6.4. A filter $F \subseteq \mathcal{D}$ in a distributive lattice $\mathcal{D}$ is *prime* if it is proper ($0 \notin F$) and $b \vee b' \in F$ implies $b \in F$ or $b' \in F$. Equivalently, just if the corresponding $\wedge$-semilattice homomorphism $f_F : \mathcal{B} \to \mathbb{2}$ is a *lattice* homomorphism.

Now if $\mathcal{B}$ is Boolean, it follows from Lemma 2.6.3 that prime filters $F \subseteq \mathcal{B}$ are in bijection with Boolean homomorphisms $\mathcal{B} \to \mathbb{2}$, via the assignment $F \mapsto f_F : \mathcal{B} \to \mathbb{2}$ with $f_F(b) = 1 \Leftrightarrow b \in F$ and $(f : \mathcal{B} \to \mathbb{2}) \mapsto F_f := f^{-1}(1) \subseteq \mathcal{B}$,

$$\mathsf{BA}(\mathcal{B}, \mathbb{2}) \;\cong\; \mathsf{PrFilters}(\mathcal{B})\,. \tag{2.21}$$

The homomorphism $f_F : \mathcal{B} \to \mathbb{2}$ may be called the *classifying map* of the prime filter $F \subseteq \mathcal{B}$. The prime filter F_f may be called the *(filter)-kernel* (or 1-*kernel*) of the homomorphism $f : \mathcal{B} \to \mathbb{2}$.

Proposition 2.6.5. *In a Boolean algebra $\mathcal{B}$, the following conditions on a filter $F \subseteq \mathcal{B}$ are equivalent.*

1. *F is prime,*
2. *the complement $\mathcal{B} \backslash F$ is a* prime ideal *(defined as a prime filter in $\mathcal{B}^{\mathsf{op}}$),*
3. *the complement $\mathcal{B} \backslash F$ is an* ideal *(defined as a filter in $\mathcal{B}^{\mathsf{op}}$),*
4. *for each $b \in \mathcal{B}$, either $b \in F$ or $\neg b \in \mathcal{F}$ and not both,*
5. *F is* maximal*: if $F \subseteq G$ and G is a filter, then $F = G$ (also called an* ultrafilter*),*
6. *the map $f_F : \mathcal{B} \to \mathbb{2}$ given by $f_F(b) = 1 \Leftrightarrow b \in F$ (as in* (2.19)*) is a Boolean homomorphism.*

Proof. Exercise! □

The following lemma is sometimes referred to as the *(Boolean) prime ideal theorem.*

Lemma 2.6.6. *Let $\mathcal{B}$ be a Boolean algebra, $I \subseteq \mathcal{B}$ an ideal, and $F \subseteq \mathcal{B}$ a filter, with $I \cap F = \emptyset$. There is a prime filter $P \supseteq F$ with $I \cap P = \emptyset$.*

Proof. Suppose first that $I = \{0\}$ is the trivial ideal, and that $\mathcal{B}$ is countable, with $b_0, b_1, ...$ an enumeration of its elements. As in the proof of the Model Existence Lemma, we build an increasing sequence of filters $F_0 \subseteq F_1 \subseteq \dots$ as follows:

$$\begin{aligned} F_0 &= F \\ F_{n+1} &= \begin{cases} F_n & \text{if } \neg b_n \in F_n \\ \{f \wedge b \,|\, f \in F_n,\ b_n \leq b\} & \text{otherwise} \end{cases} \\ P &= \bigcup_n F_n \end{aligned}$$

One then shows that each F_n is a filter, that $I \cap F_n = \emptyset$ for all n and so $I \cap P = \emptyset$, and that for each b_n, either $b_n \in P$ or $\neg b_n \in P$, whence P is prime.

For $I \subseteq \mathcal{B}$ a nontrivial ideal we take the quotient Boolean algebra $\mathcal{B} \twoheadrightarrow \mathcal{B}/I$, defined as the algebra of equivalence classes $[b]$ where $a \sim_I b \Leftrightarrow a \vee i = b \vee j$ for some $i, j \in I$. One shows that this is indeed a Boolean algebra and that the projection onto equivalence classes $\pi_I : \mathcal{B} \twoheadrightarrow \mathcal{B}/I$ is a Boolean homomorphism with (ideal) kernel $\pi^{-1}([0]) = I$. Now apply the foregoing argument to obtain a prime filter $P : \mathcal{B}/I \to \mathbb{2}$. The composite $p_I = P \circ \pi_I : \mathcal{B} \to \mathbb{2}$ is then a Boolean homomorphism with (filter) kernel $p_I^{-1}(1)$ which is prime, contains F and is disjoint from I.

The case where $\mathcal{B}$ is uncountable is left as an exercise. □

Exercise 2.6.7. Finish the proof of Lemma 2.6.6 by (i) verifying the construction of the quotient Boolean algebra $\mathcal{B} \twoheadrightarrow \mathcal{B}/I$, and (ii) considering the case where $\mathcal{B}$ is uncountable (*Hint*: either use Zorn's lemma, or well-order $\mathcal{B}$.)

Theorem 2.6.8 (Stone representation theorem)**.** *Let $\mathcal{B}$ be a Boolean algebra. There is an injective Boolean homomorphism $\mathcal{B} \rightarrowtail \mathcal{P}X_{\mathcal{B}}$ into a powerset.*

Proof. We take $X_{\mathcal{B}} = \mathsf{PrFilters}(\mathcal{B})$, the set of prime filters in $\mathcal{B}$, and consider the map $h : \mathcal{B} \to \mathcal{P}X_{\mathcal{B}}$ given by $h(b) = \{F \,|\, b \in F\}$. Clearly $h(0) = \emptyset$ and $h(1) = X$. Moreover, for *any* filter F, we have $b \in F$ and $b' \in F$ if and only if $b \wedge b' \in F$, so $h(b \wedge b') = h(b) \cap h(b')$. If F is prime, then $b \in F$ or $b' \in F$ if and only if $b \vee b' \in F$, so $h(b \vee b') = h(b) \cup h(b')$. Thus h is a Boolean homomorphism. Let $a \neq b \in \mathcal{B}$, and we want to show that $h(a) \neq h(b)$. It suffices to assume that $a < b$ (otherwise, consider $a \wedge b$, for which we cannot have both $a \wedge b = a$ and $a \wedge b = b$). We seek a prime filter $P \subseteq \mathcal{B}$ with $b \in P$ but $a \notin P$. Apply Lemma 2.6.6 to the ideal $\downarrow a$ and the filter $\uparrow b$. □

2.7 Stone duality

Note that in the Stone representation $\mathcal{B} \rightarrowtail \mathcal{P}(X_{\mathcal{B}})$ with $X_{\mathcal{B}}$ the set of prime filters in B, the powerset Boolean algebra

$$\mathcal{P}(X_{\mathcal{B}}) \cong \mathsf{Set}\big(\mathsf{BA}(\mathcal{B}, \mathbb{2}), 2\big)$$

is evidently (covariantly) functorial in $\mathcal{B}$, and has an apparent "double-dual" form $\mathcal{B}^{**}$, where $(-)^* = \mathsf{Hom}(-, 2)$, in the respective category. This suggests a possible duality between the categories BA and Set,

$$\begin{tikzcd}
\mathsf{BA}^{\mathsf{op}} \arrow[r, bend left, "*"] & \mathsf{Set} \arrow[l, bend left, "*"]
\end{tikzcd} \tag{2.22}$$

with contravariant functors

$$\mathcal{B}^* = \mathsf{BA}(\mathcal{B}, \mathbb{2}),$$

the set of prime filters of the Boolean algebra $\mathcal{B}$, and

$$S^* = \mathsf{Set}(S, 2),$$

the powerset Boolean algebra $\mathcal{P}S$ of the set S. This indeed gives a contravariant adjunction "on the right",

$$\begin{array}{cc} \mathcal{B} \to \mathcal{P}S & \mathsf{BA} \\ \hline\hline S \to X_{\mathcal{B}} & \mathsf{Set} \end{array} \tag{2.23}$$

by applying the corresponding contravariant functors

$$\begin{aligned} \mathcal{P}S &= \mathsf{Set}(S, 2), \\ X_{\mathcal{B}} &= \mathsf{BA}(\mathcal{B}, \mathbb{2}), \end{aligned}$$

and then precomposing with the respective "evaluation" natural transformations,

$$\begin{aligned} \eta_{\mathcal{B}} : \mathcal{B} &\longrightarrow \mathcal{P}(X_{\mathcal{B}}) \cong \mathsf{Set}\big(\mathsf{BA}(\mathcal{B}, \mathbb{2}), 2\big), \\ \varepsilon_S : S &\longrightarrow X_{\mathcal{P}S} \cong \mathsf{BA}\big(\mathsf{Set}(S, 2), \mathbb{2}\big). \end{aligned}$$

The homomorphism $\eta_{\mathcal{B}}$ takes an element $b \in \mathcal{B}$ to (the characteristic function of) the set of (characteristic functions of) prime filters that contain it, and the function ε_S takes an element $s \in S$ to (the characteristic function of) the principal filter $\uparrow\{s\} \subseteq \mathcal{P}S$, which is prime since the singleton set $\{s\}$ is an *atom* in $\mathcal{P}S$, i.e., a minimal, non-zero element.

Exercise 2.7.1. Verify the adjunction (2.22).

The adjunction (2.22) is not an equivalence of categories, however, because neither of the units $\eta_{\mathcal{B}}$ nor ε_S is in general an isomorphism. (Recall that a right adjoint is full and faithful just if the counit is an iso, and an equivalence if both the unit and the counit are isos.) We can improve the adjunction (2.22) by *topologizing* the set $X_{\mathcal{B}}$ of prime filters, in order to be able to cut down the powerset $\mathcal{P}(X_{\mathcal{B}}) \cong \mathsf{Set}(X_{\mathcal{B}}, 2)$ from *all* functions to just the *continuous* functions into the discrete space 2, which will then correspond to the *clopen* sets in $X_{\mathcal{B}}$.

To do this, we take as *basic open sets* of $X_{\mathcal{B}}$ all those subsets of the form:

$$B_b = \eta_{\mathcal{B}}(b) = \{P \in X_{\mathcal{B}} \mid b \in P\}, \qquad b \in \mathcal{B}. \tag{2.24}$$

These sets are closed under finite intersections, because $B_a \cap B_b = B_{a\wedge b}$. Indeed, if $P \in B_a \cap B_b$ then $a \in P$ and $b \in P$, whence $a \wedge b \in P$, and conversely (after all, $\eta_{\mathcal{B}}$ is a Boolean homomorphism!). Thus the family $(B_b)_{b\in\mathcal{B}}$ is a basis of open sets for a topology on $X_{\mathcal{B}}$.

Definition 2.7.2. For any Boolean algebra $\mathcal{B}$, the *prime spectrum* of $\mathcal{B}$ is a topological space $X_{\mathcal{B}}$ with the prime filters $P \subseteq \mathcal{B}$ as points, and the sets B_b of (2.24), for all $b \in \mathcal{B}$, as basic open sets. The prime spectrum $X_{\mathcal{B}}$ is also called the *Stone space* of $\mathcal{B}$.

Proposition 2.7.3. *The open sets $\mathcal{O}(X_{\mathcal{B}})$ of the Stone space are in order-preserving, bijective correspondence with the ideals $I \subseteq \mathcal{B}$ of the Boolean algebra, whereby the principal ideals $\downarrow b$ correspond exactly to the clopen sets B_b.*

Proof. Exercise! □

We now have an improved adjunction

$$\begin{tikzcd}
\mathsf{BA}^{\mathsf{op}} \arrow[r, bend left, "\mathsf{Spec}"] & \mathsf{Top} \arrow[l, bend left, "\mathsf{Clop}"]
\end{tikzcd} \tag{2.25}$$

$$\begin{aligned}
\mathsf{Spec}(\mathcal{B}) &= (X_{\mathcal{B}}, \mathcal{O}(X_{\mathcal{B}})) \\
\mathsf{Clop}(X) &= \mathsf{Top}(X, 2),
\end{aligned}$$

for which, up to isomorphism, the space $\mathsf{Spec}(\mathcal{B})$ has the underlying set $\mathsf{BA}(\mathcal{B}, \mathbb{2})$ given by "homming" into the Boolean algebra $\mathbb{2}$, and the Boolean algebra $\mathsf{Clop}(X) = \mathsf{Top}(X, 2)$ is similarly determined by mapping into the "topological Boolean algebra" given by the discrete space 2. Such an adjunction is said to be induced by a *dualizing object*: an object that can be regarded as "living in two different categories". Here the dualizing object 2 is acting both as a space and as a Boolean algebra. In the Lawvere duality of Chapter 1, the role of dualizing object was played by the category Set of all sets!

Now if $\eta_{\mathcal{B}} : \mathcal{B} \cong \mathsf{ClopSpec}(\mathcal{B})$, it would follow that the functor $\mathsf{Spec} : \mathsf{BA}^{\mathsf{op}} \to \mathsf{Top}$ is full and faithful. So if we then cut down the improved adjunction (2.25) to just the spaces in the image of Spec, we will obtain a "duality" (a contraviant equivalence). Toward that end, observe first that the Stone space $X_{\mathcal{B}}$ of a Boolean algebra $\mathcal{B}$ is a subspace of a product of finite discrete spaces,

$$X_{\mathcal{B}} \cong \mathsf{BA}(\mathcal{B}, \mathbb{2}) \hookrightarrow \prod_{|\mathcal{B}|} 2,$$

and is therefore a compact Hausdorff space, by Tychonoff's theorem. Indeed, the basis (2.24) is just the subspace topology on $X_{\mathcal{B}}$ with respect to the product topology on $\prod_{|\mathcal{B}|} 2$. The latter space is moreover *totally disconnected*, meaning that it has a subbasis of clopen subsets, namely all those of the form $f^{-1}(\delta) \subseteq |\mathcal{B}|$ for $f : |\mathcal{B}| \to 2$ and $\delta = 0, 1$.

Lemma 2.7.4. *The prime spectrum $X_{\mathcal{B}}$ of a Boolean algebra $\mathcal{B}$ is a totally disconnected, compact, Hausdorff space.*

Proof. Since $\prod_{|\mathcal{B}|} 2$ has just been shown to be a totally disconnected, compact Hausdorff space, we need only see that the subspace $X_{\mathcal{B}}$ is closed. Consider the subspaces

$$2^{|\mathcal{B}|}_{\wedge},\ 2^{|\mathcal{B}|}_{\vee},\ 2^{|\mathcal{B}|}_{1},\ 2^{|\mathcal{B}|}_{0} \subseteq 2^{|\mathcal{B}|}$$

consisting of the functions $f : |\mathcal{B}| \to 2$ that preserve $\wedge, \vee, 1, 0$ respectively. Since each of these is closed, so is their intersection $X_{\mathcal{B}}$. In more detail, the set of maps $f : |\mathcal{B}| \to 2$ that preserve e.g. $\wedge$ can be described as an equalizer

$$\begin{tikzcd}
2^{|\mathcal{B}|}_{\wedge} \arrow[r, tail] & 2^{|\mathcal{B}|} \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & 2^{|\mathcal{B}|\times|\mathcal{B}|}
\end{tikzcd}$$

where the maps s, t take an arrow $f : |\mathcal{B}| \to 2$ to the two different composites around the square

$$\begin{tikzcd}
{|\mathcal{B}| \times |\mathcal{B}|} \arrow[r, "\wedge"] \arrow[d, "f \times f"'] & {|\mathcal{B}|} \arrow[d, "f"] \\
2 \times 2 \arrow[r, "\wedge"'] & 2.
\end{tikzcd}$$

But the equalizer $2^{|\mathcal{B}|}_{\wedge} \rightarrowtail 2^{|\mathcal{B}|}$ is the pullback of the diagonal on $2^{|\mathcal{B}|\times|\mathcal{B}|}$, which is closed since $2^{|\mathcal{B}|\times|\mathcal{B}|}$ is Hausdorff. The other cases are analogous. □

Definition 2.7.5. A topological space is called *Stone* if it is totally disconnected, compact, and Hausdorff. Let $\mathsf{Stone} \hookrightarrow \mathsf{Top}$ be the full subcategory of topological spaces consisting of Stone spaces and continuous functions between them.

Now in order to cut down the adjunction (2.25) to a duality, we can restrict it on the topological side to just the Stone spaces, since we know this subcategory will contain the image of the functor Spec. In fact, up to isomorphism, this is exactly the image:

Theorem 2.7.6. *There is a contravariant equivalence of categories between* BA *and* Stone*,*

$$\begin{tikzcd}
\mathsf{BA}^{\mathsf{op}} \arrow[r, bend left, "*"] & \mathsf{Stone}, \arrow[l, bend left, "*"]
\end{tikzcd}$$

with contravariant functors $\mathcal{B}^* = X_{\mathcal{B}}$ the Stone space of a Boolean algebra $\mathcal{B}$, as in Definition 2.7.2, and $X^* = \mathsf{clopen}(X)$, the Boolean algebra of all clopen sets in the Stone space X.

Proof. We just need to show that the two units of the adjunction

$$\eta_{\mathcal{B}} : \mathcal{B} \to \mathsf{Top}\big(\mathsf{BA}(\mathcal{B}, \mathbb{2}), 2\big),$$
$$\varepsilon_S : S \to \mathsf{BA}\big(\mathsf{Top}(S, 2), \mathbb{2}\big).$$

are isomorphisms, the second assuming S is a Stone space.

We know by the Stone representation theorem 2.6.8 that $\eta_{\mathcal{B}}$ is an injective Boolean homomorphism, so its image, say

$$\mathcal{B}' \subseteq \mathsf{Top}\big(\mathsf{BA}(\mathcal{B}, \mathbb{2}), 2\big) \cong \mathsf{Clop}(X_{\mathcal{B}})\,,$$

is a sub-Boolean algebra of the clopen sets of $X_{\mathcal{B}}$. It suffices to show that every clopen set of $X_{\mathcal{B}}$ is in $\mathcal{B}'$. Thus let $K \subseteq X_{\mathcal{B}}$ be clopen, and take $K = \bigcup_i B_i$ a cover by basic opens B_i, all of which, note, are of the form (2.24), and so are in $\mathcal{B}'$. Since K is closed and $X_{\mathcal{B}}$ compact, K is also compact, so there is a finite subcover, each element of which is in $\mathcal{B}'$. Thus their finite union K is also in $\mathcal{B}'$.

Now let S be a Stone space and consider the continuous function

$$\varepsilon_S : S \to \mathsf{BA}\big(\mathsf{Top}(S, 2), \mathbb{2}\big) \cong X_{\mathsf{Clop}(S)}$$

which takes $s \in S$ to the prime filter $F_s = \{K \in \mathsf{Clop}(S) \mid s \in K\}$ of all clopen sets containing it. Since S is Hausdorff, ε_S is a bijection on points, and it is continuous by construction. To see that it is open, let $K \subseteq S$ be a basic clopen set. The complement $S \setminus K$ is therefore closed, and thus compact, and so is its image $\varepsilon_S(S \setminus K)$, which is therefore closed. But since ε_S is a bijection, $\varepsilon_S(S \setminus K)$ is the complement of $\varepsilon_S(K)$, which is therefore open. □

Remark 2.7.7. Another way to cut down the adjunction (2.22),

$$\begin{tikzcd}
\mathsf{BA}^{\mathsf{op}} \arrow[r, bend left, "*"] & \mathsf{Set} \arrow[l, bend left, "*"]
\end{tikzcd}$$

to an equivalence is to restrict the Boolean algebra side to the *complete, atomic* Boolean algebras $\mathsf{BA}_{\mathrm{ca}}$ and continuous (i.e. $\bigvee$-preserving) homomorphisms between them. One then obtains a duality

$$\mathsf{BA}_{\mathrm{ca}}^{\mathsf{op}} \simeq \mathsf{Set},$$

between complete, atomic Boolean algebras and sets (see Johnstone [Joh82]).

Remark 2.7.8. See Johnstone [Joh82] for a more detailed presentation of the material in this section (and much more). Also see [MR95] for a generalization to distributive lattices and Heyting algebras, as well as to "Boolean algebras with operators", i.e. algebraic models of modal logic. For more on logical duality see [Awo21]

2.8 Cartesian closed posets

We can relax the Boolean condition $\neg\neg b = b$ in order to generalize some of our results to other systems of propositional logic, represented by structured poset categories. This will be useful when we consider the "proof-relevant" versions of these as proper (*i.e.* non-poset) categories arising from systems of type theory. We begin with a basic system without the coproducts $\bot$ or $\phi \vee \psi$, and thus also without negation $\neg\phi$, which we shall therefore call the *positive propositional calculus* (a non-standard designation).

Positive propositional calculus Classically, implication $\phi \Rightarrow \psi$ can be defined by $\neg\phi \vee \psi$, but in categorical logic we prefer to consider $\phi \Rightarrow \psi$ as an *exponential*, of ψ by ϕ, defined as right adjoint to the conjunction $(-) \wedge \phi$. Since this makes sense without negation $\neg\phi$ or joins $\phi \vee \psi$, we can study just the cartesian closed fragment separately, and then add those other operations later. The same approach will be used for type theory in Chapter **??**.

Definition 2.8.1. The *positive propositional calculus* PPC is the subsystem of the propositional calculus of Section 2.1 containing just (finite) conjunction and implication. So PPC is the set of all propositional formulas ϕ constructed from propositional variables $p_1, p_2, ...,$ a constant $\top$ for truth, and binary connectives for conjunction $\phi \wedge \psi$, and implication $\phi \Rightarrow \psi$.

As a category, PPC is a preorder under the relation $\phi \vdash \psi$ of logical entailment, determined, say, by the natural deduction system of section 2.1 . As usual, it will be convenient to pass to the poset reflection of the preorder, which we shall denote by

$$\mathcal{C}_{\mathsf{PPC}}$$

by identifying ϕ and ψ when $\phi \dashv\vdash \psi$. (This is the (syntactic) *Lindenbaum-Tarski* algebra of the system PPC of positive propositional logic, as in Section 2.5.)

The conjunction $\phi \wedge \psi$ is a greatest lower bound of ϕ and ψ in $\mathcal{C}_{\mathsf{PPC}}$, because $\phi \wedge \psi \vdash \phi$ and $\phi \wedge \psi \vdash \psi$, and for all ϑ, if $\vartheta \vdash \phi$ and $\vartheta \vdash \psi$ then $\vartheta \vdash \phi \wedge \psi$. Since binary products in a poset are the same thing as greatest lower bounds, we see that $\mathcal{C}_{\mathsf{PPC}}$ has all binary products; and of course $\top$ is a terminal object, so $\mathcal{C}_{\mathsf{PPC}}$ is a $\wedge$-semilattice.

We have already remarked that implication is right adjoint to conjunction in the sense that for any ϕ,

$$(-) \wedge \phi \;\dashv\; \phi \Rightarrow (-)\ . \tag{2.26}$$

Therefore $\phi \Rightarrow \psi$ is an exponential in $\mathcal{C}_{\mathsf{PPC}}$. The counit of the adjunction (the "evaluation" arrow) is the entailment

$$(\phi \Rightarrow \psi) \wedge \phi \vdash \psi\ ,$$

i.e. the familiar logical rule of *modus ponens.*

We therefore have the following:

Proposition 2.8.2. *The poset $\mathcal{C}_{\mathsf{PPC}}$ of positive propositional calculus is cartesian closed.*

We can use this fact to show that the positive propositional calculus is *deductively complete* with respect to the following notion of *Kripke semantics* [?].

Definition 2.8.3 (Kripke semantics)**.** 1. A *Kripke model* is a poset K (the *worlds*) equipped with a relation

$$k \Vdash p$$

between elements $k \in K$ and propositional variables p, such that for all $j \in K$,

$$j \leq k,\ k \Vdash p \quad \text{implies} \quad j \Vdash p\,. \tag{2.27}$$

2. Given a Kripke model $(K, \Vdash)$, extend the relation $\Vdash$ to all formulas ϕ in PPC by defining the relation of *holding in a world* $k \in K$ inductively by the following conditions:

$$\begin{array}{lll} k \Vdash \top & \text{always,} & \\ k \Vdash \phi \wedge \psi & \text{iff} & k \Vdash \phi \text{ and } k \Vdash \psi\,, \\ k \Vdash \phi \Rightarrow \psi & \text{iff} & \text{for all } j \leq k, \text{ if } j \Vdash \phi, \text{ then } j \Vdash \psi\,. \end{array} \tag{2.28}$$

3. Finally, say that ϕ *holds in the Kripke model* $(K, \Vdash)$, written

$$K \Vdash \phi$$

if $k \Vdash \phi$ for all $k \in K$. (One sometimes also says that ϕ *holds on* the poset K if $K \Vdash \phi$ for all such Kripke relations $\Vdash$ on K.)

Theorem 2.8.4 (Kripke completeness for PPC)**.** *A propositional formula ϕ is provable from the rules of deduction for PPC if, and only if, $K \Vdash \phi$ for all Kripke models $(K, \Vdash)$,*

$$\mathsf{PPC} \vdash \phi \qquad \textit{iff} \qquad K \Vdash \phi \quad \textit{for all } (K, \Vdash).$$

For the proof, we first require the following, which generalizes the discussion around (2.19) in Section 2.6.

Lemma 2.8.5. *For any poset P, the poset $\mathsf{Down}(P)$ of all downsets in P, ordered by inclusion, is cartesian closed. Moreover, the downset embedding,*

$$\downarrow(-) : P \longrightarrow \mathsf{Down}(P)$$

preserves any CCC structure that exists in P.

Proof. The total downset P is obviously terminal, and for any downsets $S, T \in \mathsf{Down}(P)$, the intersection $S \cap T$ is also closed down, so we have the products $S \wedge T = S \cap T$. For the exponential, set

$$S \Rightarrow T = \{p \in P \mid \downarrow(p) \cap S \subseteq T\}. \tag{2.29}$$

Then for any downset Q we have

$$\begin{array}{lll} Q \subseteq S \Rightarrow T & \text{iff} & \text{for all } q \in Q, \quad q \in S \Rightarrow T\,, \\ & \text{iff} & \text{for all } q \in Q, \quad \downarrow(q) \cap S \subseteq T\,, \\ & \text{iff} & \bigcup_{q \in Q}(\downarrow(q) \cap S) \subseteq T\,, \\ & \text{iff} & (\bigcup_{q \in Q} \downarrow(q)) \cap S \subseteq T\,, \\ & \text{iff} & Q \cap S \subseteq T\,. \end{array}$$

The preservation of CCC structure by $\downarrow(-) : P \longrightarrow \mathsf{Down}(P)$ follows from its preservation by the Yoneda embedding, of which we know $\downarrow(-)$ to be a factor,

$$\begin{tikzcd}
 & \mathsf{Set}^{P^{\mathsf{op}}} \\
P \arrow[ur, hook, "\mathsf{y}"] \arrow[r, "\downarrow(-)"'] & \mathsf{Down}(P) \arrow[u, hook]
\end{tikzcd}$$

Indeed, we can identify $\mathsf{Down}(P)$ with the subcategory $\mathsf{Sub}(1)$ of subobjects of 1 in $\mathsf{Set}^{P^{\mathsf{op}}}$ and the result follows by using the left adjoint left inverse sup of the inclusion

$$\mathsf{sup} \dashv i : \mathsf{Sub}(1) \hookrightarrow \mathsf{Set}^{P^{\mathsf{op}}} ,$$

to be considered later (*cf.* Lemma 4.7.1).

But it is also easy enough to check it directly: Preservation of any limits 1, $p \wedge q$ that exist in P are clear, since these are pointwise. Then suppose $p \Rightarrow q$ is an exponential; so for any downset D we have:

$$\begin{array}{rll}
D \subseteq \downarrow(p \Rightarrow q) & \text{iff} & d \in \downarrow(p \Rightarrow q)\ , \text{ for all } d \in D \\
 & \text{iff} & d \leq p \Rightarrow q\ , \text{ for all } d \in D \\
 & \text{iff} & d \wedge p \leq q\ , \text{ for all } d \in D \\
 & \text{iff} & \downarrow(d \wedge p) \subseteq \downarrow(q)\ , \text{ for all } d \in D \\
 & \text{iff} & \downarrow(d) \cap \downarrow(p) \subseteq \downarrow(q)\ , \text{ for all } d \in D \\
 & \text{iff} & D \subseteq \downarrow(p) \Rightarrow \downarrow(q)
\end{array}$$

where the last line is by (2.29). Now take D to be $\downarrow(p \Rightarrow q)$ and $\downarrow(p) \Rightarrow \downarrow(q)$ respectively (or just apply Yoneda!). (Note that in line (3) we assumed that $d \wedge p$ exists for all $d \in D$; this can be avoided by a slightly more complicated argument.) □

Proof. (of Theorem 2.8.4) The proof follows a now-familiar pattern, which we only sketch:

1. The syntactic category $\mathcal{C}_{\mathsf{PPC}}$ is a CCC, with $\top = 1$, $\phi \times \psi = \phi \wedge \psi$, and $\psi^{\phi} = \phi \Rightarrow \psi$. In fact, it is the free cartesian closed poset on the generating set $\mathsf{Var} = \{p_1, p_2, \dots\}$ of propositional variables.

2. A (Kripke) model $(K, \Vdash)$ is the same thing as a CCC functor $\mathcal{C}_{\mathsf{PPC}} \to \mathsf{Down}(K)$, which by Step 1 is just an arbitrary map $\mathsf{Var} \to \mathsf{Down}(K)$, as in (2.27). To see this, observe that we have a bijective correspondence between CCC functors $[\![-]\!]$ and Kripke relations $\Vdash$; indeed, by the exponential adjunction in the cartesian closed category Pos, there is a natural bijection,

$$\begin{array}{c}
\Vdash : K^{\mathsf{op}} \times \mathcal{C}_{\mathsf{PPC}} \longrightarrow 2 \\
\hline\hline
[\![-]\!] : \mathcal{C}_{\mathsf{PPC}} \longrightarrow 2^{K^{\mathsf{op}}} \cong \mathsf{Down}(K)
\end{array}$$

where we use the poset $\mathbb{2}$ to classify downsets in a poset K via upsets in K^{op},

$$\mathbb{2}^{P^{\mathsf{op}}} \cong \mathsf{Pos}(K^{\mathsf{op}}, \mathbb{2}) \cong \mathsf{Down}(K)\,,$$

by taking the 1-kernel $f^{-1}(1) \subseteq K$ of a monotone map $f : K^{\mathsf{op}} \to \mathbb{2}$. (The contravariance will be convenient in Step 3). Note that the monotonicity of $\Vdash$ yields the conditions

$$j \leq k\,,\ k \Vdash \phi \implies j \Vdash \phi$$

and

$$k \Vdash \phi\,,\ \phi \vdash \psi \implies k \Vdash \psi\,.$$

And the CCC preservation of the transpose $[\![-]\!]$ yields the Kripke forcing conditions (2.28) (exercise!).

3. For any model $(K, \Vdash)$, by the adjunction in (2) we then have

$$K \Vdash \phi \iff [\![\phi]\!] = K\,,$$

with $K \subseteq K$ the maximal downset.

4. Because the downset/Yoneda embedding $\downarrow$ preserves the CCC structure (by Lemma 2.8.5), $\mathcal{C}_{\mathsf{PPC}}$ has a *canonical model*, namely the special case of (2) with $K = \mathcal{C}_{\mathsf{PPC}}$ and $\Vdash$ resulting from the trasposition:

$$\begin{array}{c} \downarrow(-)\ :\ \mathcal{C}_{\mathsf{PPC}} \longrightarrow \mathsf{Down}(\mathcal{C}_{\mathsf{PPC}}) \cong \mathbb{2}^{\mathcal{C}^{\mathsf{op}}_{\mathsf{PPC}}} \\ \hline\hline \Vdash\ :\ \mathcal{C}^{\mathsf{op}}_{\mathsf{PPC}} \times \mathcal{C}_{\mathsf{PPC}} \longrightarrow \mathbb{2} \end{array}$$

5. Now note that for the Kripke relation $\Vdash$ in (4), we have $\Vdash\ =\ \vdash$ since it's just the transpose of the Yoneda embedding, and the poset $\mathcal{C}_{\mathsf{PPC}}$ is ordered by $\phi \vdash \psi$. So the canonical model is *logically generic*, in the sense that

$$\phi \Vdash \psi \quad \iff \quad \phi \vdash \psi\,,$$

and so in particular,

$$\mathcal{C}_{\mathsf{PPC}} \Vdash \phi \quad \iff \quad \mathsf{PPC} \vdash \phi\,.$$

□

Exercise 2.8.6. Verify the claim in (2) that CCC preservation of the transpose $[\![-]\!]$ of $\Vdash$ yields the Kripke forcing conditions (2.28).

Exercise 2.8.7. Give a Kripke countermodel to show that $\mathsf{PPC} \nvdash (\phi \Rightarrow \psi) \Rightarrow \phi$.

2.9 Heyting algebras

Let us now extend the positive propositional calculus to the full intuitionistic propositional calculus. This involves adding the finite coproducts 0 and $p \vee q$ to the notion of a cartesian closed poset, to arrive at the general notion of a Heyting algebra. Heyting algebras are to intuitionistic logic as Boolean algebras are to classical logic: each is an algebraic description of the corresponding logical calculus. We shall review both the algebraic and the logical points of view; as we shall see, many aspects of the theory of Boolean algebras carry over to Heyting algebras. For instance, in order to prove the Kripke completeness of the full system of intuitionistic propositional calculus, we will need an alternative to Lemma 2.8.5, because the Yoneda embedding does not in general preserve coproducts. For that we will again use a version of the Stone representation theorem, this time in a generalized form due to Joyal.

Distributive lattices

Recall first that a (bounded) *lattice* is a poset that has finite limits and colimits. In other words, a lattice $(L, \leq, \wedge, \vee, 1, 0)$ is a poset $(L, \leq)$ with distinguished elements $1, 0 \in L$, and binary operations of meet $\wedge$ and join $\vee$, satisfying for all $x, y, z \in L$,

$$0 \leq x \leq 1 \qquad\qquad \frac{z \leq x \qquad z \leq y}{\overline{z \leq x \wedge y}} \qquad\qquad \frac{x \leq z \qquad y \leq z}{\overline{x \vee y \leq z}}$$

A *lattice homomorphism* is a function $f : L \to K$ between lattices which preserves finite limits and colimits, i.e., $f0 = 0$, $f1 = 1$, $f(x \wedge y) = fx \wedge fy$, and $f(x \vee y) = fx \vee fy$. The category of lattices and lattice homomorphisms is denoted by Lat.

Lattices are an algebraic theory, and can be axiomatized equationally in a signature with two distinguished elements 0 and 1 and two binary operations $\wedge$ and $\vee$, satisfying the following equations:

$$\begin{aligned}
(x \wedge y) \wedge z &= x \wedge (y \wedge z)\ , & (x \vee y) \vee z &= x \vee (y \vee z)\ ,\\
x \wedge y &= y \wedge x\ , & x \vee y &= y \vee x\ ,\\
x \wedge x &= x\ , & x \vee x &= x\ ,\\
1 \wedge x &= x\ , & 0 \vee x &= x\ ,\\
&x \wedge (y \vee x) = x = (x \wedge y) \vee x\ .
\end{aligned} \tag{2.30}$$

The partial order on L is then determined by

$$x \leq y \iff x = x \wedge y\ .$$

Exercise 2.9.1. Show that in a lattice we also have $x \leq y$ if and only if $x \vee y = y$.

A lattice is *distributive* if the following distributive laws hold:

$$\begin{aligned} (x \vee y) \wedge z &= (x \wedge z) \vee (y \wedge z)\ , \\ (x \wedge y) \vee z &= (x \vee z) \wedge (y \vee z)\ . \end{aligned} \tag{2.31}$$

It turns out that if one distributive law holds then so does the other [Joh82, I.1.5].

Definition 2.9.2. A *Heyting algebra* is a cartesian closed lattice. This means that a Heyting algebra $\mathcal{H}$ has a binary operation of *implication* $x \Rightarrow y$, satisfying the following condition, for all $x, y, z \in \mathcal{H}$:

$$\frac{z \leq x \Rightarrow y}{\overline{z \wedge x \leq y}}$$

A *Heyting algebra homomorphism* is a lattice homomorphism $f : \mathcal{K} \to \mathcal{H}$ between Heyting algebras that preserves implication, i.e., $f(x \Rightarrow y) = (fx \Rightarrow fy)$. The category of Heyting algebras and their homomorphisms is denoted by Heyt. (*Caution*: unlike Boolean algebras, the subcategory of lattices consisting of Heyting algebras and their homomorphisms is *not full.*)

Heyting algebras can be axiomatized equationally as a set H with two distinguished elements 0 and 1 and three binary operations $\wedge$, $\vee$ and $\Rightarrow$. The equations for a Heyting algebra are the ones listed in (2.30), as well as the following ones for $\Rightarrow$.

$$\begin{aligned} (x \Rightarrow x) &= 1\ , \\ x \wedge (x \Rightarrow y) &= x \wedge y\ , \\ y \wedge (x \Rightarrow y) &= y\ , \\ (x \Rightarrow (y \wedge z)) &= (x \Rightarrow y) \wedge (x \Rightarrow z)\ . \end{aligned} \tag{2.32}$$

For a proof, see [Joh82, I.1], where one can also find a proof that every Heyting algebra is distributive (exercise!).

Exercise 2.9.3. Show that every Heyting algebra is indeed a distributive lattice.

Example 2.9.4. We know from Lemma 2.8.5 that for any poset P, the poset $\mathsf{Down}(P)$ of all downsets in P, ordered by inclusion, is cartesian closed. Moreover, we know that

$$\mathsf{Down}(P) \cong \mathbb{2}^{P^{\mathsf{op}}} \cong \mathsf{Pos}(P^{\mathsf{op}}, \mathbb{2})\,,$$

the latter regarded as a poset with the pointwise ordering on the monotone maps $P^{\mathsf{op}} \to \mathbb{2}$ (*i.e.* the natural transformations). The assignment takes a map $f : P^{\mathsf{op}} \to \mathbb{2}$ to the filter-kernel $f^{-1}(1) \subseteq P^{\mathsf{op}}$, which is therefore a downset in P. Indeed, if $f \leq g$ then $p \in f^{-1}(1) \iff fp = 1$ which implies $gp = 1 \iff p \in g^{-1}(1)$, so $f^{-1}(1) \subseteq g^{-1}(1)$, and these upsets in P^{op} are downsets in P.

Since $\mathbb{2}$ is a lattice, we can take joins $f \vee g$ in $\mathsf{Pos}(P^{\mathsf{op}}, \mathbb{2})$ pointwise, in order to get joins in $\mathsf{Down}(P) \cong \mathsf{Pos}(P^{\mathsf{op}}, \mathbb{2})$, which then correspond to (set theoretic) unions of the corresponding downsets $f^{-1}(1) \cup g^{-1}(1)$. Thus for any poset P, the lattice $\mathsf{Down}(P)$ is a

Heyting algebra, with the downsets ordered by inclusion, and the (contravariant) classifying maps $P^{\mathsf{op}} \to \mathbb{2}$ ordered pointwise.

Of course, one can compose the classifying maps with the negation iso $\neg : \mathbb{2} \xrightarrow{\sim} \mathbb{2}$ to get $\mathsf{Down}(P) \cong \mathsf{Pos}(P, \mathbb{2})$, with *covariant* classifying maps $P \to \mathbb{2}$ for the downsets, using the ideal-kernels $f^{-1}(0) \subseteq P$ instead; but then the ordering on $\mathsf{Pos}(P, \mathbb{2})$ will be the *reverse pointwise ordering* of maps $f : P \to \mathbb{2}$.

Intuitionistic propositional calculus

There is an obvious forgetful functor $U : \mathsf{Heyt} \to \mathsf{Set}$ mapping a Heyting algebra to its underlying set, and a homomorphism of Heyting algebras to the underlying function. Because Heyting algebras are also models of an equational theory, there is a left adjoint $H \dashv U$, which is the usual “free” construction for algebras, mapping a set S to the free Heyting algebra $H(S)$ generated by it. As for all algebraic structures, the construction of $H(S)$ can be performed in two steps: first, define a set $H[S]$ of formal expressions in the signature, and then quotient it by an equivalence relation generated by the equations.

In more detail, let $H[S]$ be the set of formal expressions generated inductively by the following rules:

1. Generators: if $x \in S$ then $x \in H[S]$.

2. Constants: $\bot, \top \in H[S]$.

3. Connectives: if $\phi, \psi \in H[S]$ then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \Rightarrow \psi) \in H[S]$.

We then impose an equivalence relation $\sim$ on $H[S]$, defined as the smallest equivalence relation containing all instances of the axioms (2.30) and (2.32) and closed under substitution of equals for equals (sometimes called the smallest *congruence*). This then forces the quotient

$$H(S) = H[S]/_{\sim}$$

to be a Heyting algebra, as is easily checked.

We define the action of the functor H on morphisms as usual: a function $f : S \to T$ is mapped to the Heyting algebra homomorphism $H(f) : H(S) \to H(T)$ (well-)defined (on equivalence classes) by

$$H(f)\bot = \bot \,, \qquad H(f)\bot = \bot \,, \qquad H(f)x = fx \,,$$
$$H(f)(\phi * \psi) = (H(f)\phi) * (H(f)\psi) \,,$$

where $*$ stands for $\wedge$, $\vee$ or $\Rightarrow$.

The inclusion of generators $\eta_S : S \to UH(S)$ into the underlying set of the free Heyting algebra $H(S)$ is then the component at S of a natural transformation $\eta : 1_{\mathsf{Set}} \Longrightarrow U \circ H$, which is of course the unit of the adjunction $H \dashv U$. To see this, consider a Heyting algebra $\mathcal{K}$ and an arbitrary function $f : S \to U\mathcal{K}$. Then the Heyting algebra homomorphism

$\overline{f} : H(S) \to \mathcal{K}$ is defined in the evident way, by

$$\overline{f}\bot = \bot \,, \qquad \overline{f}\bot = \bot \,, \qquad \overline{f}x = fx \,,$$
$$\overline{f}(\phi * \psi) = (\overline{f}\phi) * (\overline{f}\psi) \,,$$

where, again, $*$ stands for $\wedge$, $\vee$ or $\Rightarrow$. The map $\overline{f}$ then makes the following triangle in Set commute:

$$\begin{tikzcd}
S \arrow[r, "\eta_S"] \arrow[dr, "f"'] & UH(S) \arrow[d, "U\overline{f}"] \\
& U\mathcal{K}
\end{tikzcd}$$

The homomorphism $\overline{f} : H(S) \to \mathcal{K}$ is the unique one with this property, because any two homomorphisms from $H(S)$ that agree on generators must clearly be equal (formally, this can be proved by induction on the structure of the expressions in $H[S]$).

We can now define the *intuitionistic propositional calculus* IPC to be the free Heyting algebra $H(p_0, p_1, \dots)$ on countably many generators $\{p_0, p_1, \dots\}$, called *atomic propositions* or *propositional variables.* This is a somewhat unorthodox definition from a logical point of view—normally we would start from a *deductive calculus* consisting of a formal language, entailment judgements, and rules of inference. But of course, by now, we realize that the two approaches are essentially equivalent.

Having said that, let us also briefly describe IPC in the conventional way: The formulas are all those given in Section 2.1, and the rules of inference are those of the system of natural deduction from Section 2.1, but *without the classical rules.*

Then let $\mathcal{C}_{\mathsf{IPC}}$ be the poset reflection of the formulas of IPC, preordered by entailment $\phi \vdash \psi$. The elements of $\mathcal{C}_{\mathsf{IPC}}$ are thus equivalence classes $[\phi]$ of formulas, where two formulas ϕ and ψ are equivalent if both $\phi \vdash \psi$ and $\psi \vdash \phi$ are provable in natural deduction, without the classical rules,

$$[\phi] = [\psi] \iff \phi \dashv\vdash \psi \,.$$

This syntactic category $\mathcal{C}_{\mathsf{IPC}}$ is then easily seen to be the free Heyting algebra on countably many generators $\{p_0, p_1, \dots\}$,

$$\mathcal{C}_{\mathsf{IPC}} \cong H(p_0, p_1, \dots) \,,$$

just as the corresponding "Lindenbaum-Tarski" Boolean algebra $\mathcal{B}[p_0, p_1, \dots]$ was seen to be the free Boolean algebra on the propositional variables as generators.

Classical propositional calculus redux

Let us have another look back at the theory of classical propositional logic from the current point of view, *i.e.* as a special kind of Heyting algebra. An element $x \in L$ of a lattice L is said to be *complemented* when there exists $y \in L$ such that

$$x \wedge y = 0 \,, \qquad\qquad x \vee y = 1 \,.$$

We say that y is the *complement* of x. In a distributive lattice, the complement of x is unique if it exists. Indeed, if both y and z are complements of x then

$$y \wedge z = (y \wedge z) \vee 0 = (y \wedge z) \vee (y \wedge x) = y \wedge (z \vee x) = y \wedge 1 = y \ ,$$

hence $y \leq z$. A symmetric argument shows that $z \leq y$, therefore $y = z$. The complement of x, if it exists, is denoted by $\neg x$.

A *Boolean algebra* can be defined as a distributive lattice in which every element is complemented. In other words, a Boolean algebra B has a *complementation operation* $\neg : B \to B$ which satisfies, for all $x \in B$,

$$x \wedge \neg x = 0 \ , \qquad\qquad x \vee \neg x = 1 \ . \tag{2.33}$$

The full subcategory of Lat consisting of Boolean algebras is denoted by BA.

Exercise 2.9.5. Prove that every Boolean algebra is a Heyting algebra. (*Hint*: how is implication encoded in terms of negation and disjunction in classical logic?)

In a Heyting algebra, not every element is complemented. However, we can still define a *pseudo complement* or *negation* operation $\neg$ by

$$\neg x = (x \Rightarrow 0) \ ,$$

Then $\neg x$ is the largest element for which $x \wedge \neg x = 0$. While in a Boolean algebra $\neg\neg x = x$, in a Heyting algebra we only have $x \leq \neg\neg x$ in general. An element x of a Heyting algebra for which $x = \neg\neg x$ is called *regular*.

Exercise 2.9.6. Derive the following properties of negation in a *Heyting* algebra:

$$\begin{gathered} x \leq \neg\neg x \ , \\ \neg x = \neg\neg\neg x \ , \\ x \leq y \Rightarrow \neg y \leq \neg x \ , \\ \neg\neg(x \wedge y) = \neg\neg x \wedge \neg\neg y \ . \end{gathered}$$

Exercise 2.9.7. Prove that the topology $\mathcal{O}X$ of any topological space X is a Heyting algebra. Describe in topological language the implication $U \Rightarrow V$, the negation $\neg U$, and the regular elements $U = \neg\neg U$ in $\mathcal{O}X$.

Exercise 2.9.8. Show that for a Heyting algebra H, the regular elements of H form a Boolean algebra $H_{\neg\neg} = \{x \in H \mid x = \neg\neg x\}$. Here $H_{\neg\neg}$ is viewed as a subposet of H. Hint: negation $\neg'$, conjunction $\wedge'$, and disjunction $\vee'$ in $H_{\neg\neg}$ are expressed as follows in terms of negation, conjunction and disjunction in H, for $x, y \in H_{\neg\neg}$:

$$\neg' x = \neg x \ , \qquad x \wedge' y = \neg\neg(x \wedge y) \ , \qquad x \vee' y = \neg\neg(x \vee y) \ .$$

From logical point of view, the *classical propositional calculus* CPC is obtained from the intuitionistic propositional calculus by the addition of the logical law known as *tertium non datur*, or the *law of excluded middle*:

$$\frac{}{\Gamma \vdash \phi \vee \neg\phi}$$

Alternatively, we could add the rule of *reductio ad absurdum*, or *proof by contradiction*:

$$\frac{\Gamma \vdash \neg\neg\phi}{\Gamma \vdash \phi} \;.$$

Identifying logically equivalent formulas of CPC, we obtain a poset $\mathcal{C}_{\mathsf{CPC}}$ ordered by logical entailment. This poset is, of course, the *free Boolean algebra* on the countably many generators $\{p_0, p_1, \dots\}$. The free Boolean algebra can be constructed just as the free Heyting algebra above, either equationally, or in terms of deduction. The equational axioms for a Boolean algebra are the axioms for a lattice (2.30), the distributive laws (2.31), and the complement laws (2.33).

Exercise* 2.9.9. Is $\mathcal{C}_{\mathsf{CPC}}$ isomorphic to the Boolean algebra $\mathcal{C}_{\mathsf{IPC}_{\neg\neg}}$ of the regular elements of $\mathcal{C}_{\mathsf{IPC}}$?

Exercise 2.9.10. Show that in a Heyting algebra H, one has $\neg\neg x = x$ for all $x \in H$ if, and only if, $y \vee \neg y = 1$ for all $y \in H$. *Hint*: half of the equivalence is easy. For the other half, observe that the assumption $\neg\neg x = x$ means that negation is an order-reversing bijection $H \to H$. It therefore transforms joins into meets and vice versa, and so the *De Morgan laws* hold:

$$\neg(x \wedge y) = \neg x \vee \neg y \;, \qquad\qquad \neg(x \vee y) = \neg x \wedge \neg y \;.$$

Together with $y \wedge \neg y = 0$, the De Morgan laws easily imply $y \vee \neg y = 1$. See [Joh82, I.1.11].

Kripke semantics for IPC

Let us now prove the Kripke completeness of IPC, extending Theorem 2.8.4, namely:

Theorem 2.9.11 (Kripke completeness for IPC)**.** *Let $(K, \Vdash)$ be a Kripke model,* i.e. *a poset K equipped with a* forcing relation *$k \Vdash p$ between elements $k \in K$ and propositional variables p, satisfying*

$$j \leq k,\ k \Vdash p \quad \textit{implies} \quad j \Vdash p. \tag{2.34}$$

Extend $\Vdash$ to all formulas ϕ in IPC *by defining*

$$\begin{array}{lll} k \Vdash \top & \textit{always,} & \\ k \Vdash \bot & \textit{never,} & \\ k \Vdash \phi \wedge \psi & \textit{iff} & k \Vdash \phi \textit{ and } k \Vdash \psi \,, \qquad (2.35)\\ k \Vdash \phi \vee \psi & \textit{iff} & k \Vdash \phi \textit{ or } k \Vdash \psi \,, \qquad (2.36)\\ k \Vdash \phi \Rightarrow \psi & \textit{iff} & \textit{for all } j \leq k, \textit{ if } j \Vdash \phi, \textit{ then } j \Vdash \psi \,. \end{array}$$

Finally, write $K \Vdash \phi$ if $k \Vdash \phi$ for all $k \in K$.

A propositional formula ϕ is then provable from the rules of deduction for IPC *if, and only if, $K \Vdash \phi$ for all Kripke models $(K, \Vdash)$. Briefly:*

$$\mathsf{IPC} \vdash \phi \quad \textit{iff} \quad K \Vdash \phi \textit{ for all } (K, \Vdash).$$

Let us first see that we cannot simply reuse the proof from Theorem 2.8.4 for the positive fragment PPC, because the downset (Yoneda) embedding that we used there

$$\downarrow\ : \mathcal{C}_{\mathsf{PPC}} \hookrightarrow \ \mathsf{Down}(\mathcal{C}_{\mathsf{PPC}}) \tag{2.37}$$

would not preserve the coproducts $\bot$ and $\phi \vee \psi$. Indeed, $\downarrow(\bot) \neq \emptyset$, because it contains $\bot$ itself! And in general $\downarrow(\phi \vee \psi) \neq \downarrow(\phi) \cup \downarrow(\psi)$, because the righthand side need not contain, e.g., $\phi \vee \psi$.

Instead, we will generalize the Stone Representation theorem 2.6.8 from Boolean algebras to Heyting algebras, using a theorem due to A. Joyal (cf. [MR95, MH92]). First, recall that the Stone representation provided, for any Boolean algebra $\mathcal{B}$, an injective Boolean homomorphism into a powerset,

$$\mathcal{B} \rightarrowtail \mathcal{P}X\,.$$

For X we took the set of prime filters, which we identified with the homset of Boolean homomorphisms $\mathsf{BA}(\mathcal{B}, \mathbb{2})$ by taking the filter-kernel $f^{-1}(1) \subseteq \mathcal{B}$ of a homomorphism $f : \mathcal{B} \to \mathbb{2}$. The injective homomorphism $\eta : \mathcal{B} \rightarrowtail \mathcal{P}(\mathsf{BA}(\mathcal{B}, \mathbb{2}))$ was then given by:

$$\eta(b) = \{F \mid b \in F\} = \{f : \mathcal{B} \to \mathbb{2} \mid f(b) = 1\}\,.$$

Now, the set $\mathsf{BA}(\mathcal{B}, \mathbb{2})$ can be regarded as a (discrete) poset, and since the inclusion $\mathsf{Set} \hookrightarrow \mathsf{Pos}$ as discrete posets is left adjoint to the forgetful functor $|-| : \mathsf{Pos} \to \mathsf{Set}$, for the powerset $\mathcal{P}(\mathsf{BA}(\mathcal{B}, \mathbb{2}))$ we have

$$\mathcal{P}(\mathsf{BA}(\mathcal{B}, \mathbb{2})) \cong \mathsf{Set}(\mathsf{BA}(\mathcal{B}, \mathbb{2}), 2) \cong \mathsf{Pos}(\mathsf{BA}(\mathcal{B}, \mathbb{2}), \mathbb{2}) \cong \mathbb{2}^{\mathsf{BA}(\mathcal{B}, \mathbb{2})}$$

where the latter is the exponential in the cartesian closed category Pos. Transposing the composite of this iso with the Stone representation $\eta : \mathcal{B} \rightarrowtail \mathcal{P}X$ in Pos,

$$\frac{\eta : \mathcal{B} \rightarrowtail \mathcal{P}(\mathsf{BA}(\mathcal{B}, \mathbb{2})) \cong \mathbb{2}^{\mathsf{BA}(\mathcal{B}, \mathbb{2})}}{\tilde{\eta} : \mathsf{BA}(\mathcal{B}, \mathbb{2}) \times \mathcal{B} \to \mathbb{2}}$$

we arrive at the (monotone) evaluation map

$$\tilde{\eta} = \mathsf{eval} : \mathsf{BA}(\mathcal{B}, \mathbb{2}) \times \mathcal{B} \to \mathbb{2}. \tag{2.38}$$

Finally, recall that the category of Boolean algebras is full in the category DLat of distributive lattices, so that

$$\mathsf{BA}(\mathcal{B}, \mathbb{2}) = \mathsf{DLat}(\mathcal{B}, \mathbb{2})\,.$$

Now for any *Heyting algebra* $\mathcal{H}$ (or indeed any distributive lattice), the homset $\mathsf{DLat}(\mathcal{H}, \mathbb{2})$, ordered pointwise, is isomorphic to the *poset* of all prime filters in $\mathcal{H}$ ordered by inclusion, again by taking $h : \mathcal{H} \to \mathbb{2}$ to its (filter) kernel $h^{-1}\{1\} \subseteq \mathcal{H}$. In particular, when $\mathcal{H}$ is not Boolean, the poset $\mathsf{DLat}(\mathcal{H}, \mathbb{2})$ is no longer discrete, since prime filters in a Heyting algebra need not be maximal. Indeed, recall that Proposition 2.6.5 described the prime filters in a Boolean algebra $\mathcal{B}$ as those with a classifying map $f : \mathcal{B} \to \mathbb{2}$ that is a lattice homomorphism and therefore those with a complement $f^{-1}(0) \subseteq \mathcal{B}$ that is a (prime) ideal. In the Boolean case, these were also the *maximal* filters, because the preservation of Boolean negation $\neg b$ allowed us to deduce that for every $b \in \mathcal{B}$, exactly one of b or $\neg b$ must be in such a filter F. In a Heyting algebra, however, the last condition need not obtain; and indeed prime filters in a Heyting algebra need not be maximal.

The transpose in Pos of the evaluation map,

$$\mathsf{eval} : \mathsf{DLat}(\mathcal{H}, \mathbb{2}) \times \mathcal{H} \to \mathbb{2}. \tag{2.39}$$

is again a monotone map

$$\eta : \mathcal{H} \longrightarrow \mathbb{2}^{\mathsf{DLat}(\mathcal{H},\mathbb{2})}, \tag{2.40}$$

which takes $p \in \mathcal{H}$ to the "evaluation at p" map $f \mapsto f(p) \in \mathbb{2}$, i.e.,

$$\eta_p(f) = f(p) \qquad \text{for } p \in \mathcal{H} \text{ and } f : \mathcal{H} \to \mathbb{2}\,.$$

As before (cf. Example 2.9.4), the poset $\mathbb{2}^{\mathsf{DLat}(\mathcal{H},\mathbb{2})}$ (ordered pointwise) may be identified with the downsets in the poset $\mathsf{DLat}(\mathcal{H}, \mathbb{2})^{\mathsf{op}}$, ordered by inclusion, which recall from Example 2.9.4 is always a Heyting algebra. Thus, in sum, for any Heyting algebra $\mathcal{H}$, we have a monotone map,

$$\eta : \mathcal{H} \longrightarrow \mathsf{Down}(\mathsf{DLat}(\mathcal{H}, \mathbb{2})^{\mathsf{op}})\,, \tag{2.41}$$

generalizing the Stone representation from Boolean to Heyting algebras.

Theorem 2.9.12 (Joyal)**.** *Let $\mathcal{H}$ be a Heyting algebra. There is an injective homomorphism of Heyting algebras*

$$\mathcal{H} \rightarrowtail \mathsf{Down}(J)$$

into the Heyting algebra of downsets in a poset J.

Note that in this form, the theorem literally generalizes the Stone representation theorem: when $\mathcal{H}$ is Boolean we can take J to be discrete, and then $\mathsf{Down}(J) \cong \mathsf{Pos}(J, \mathbb{2}) \cong \mathsf{Set}(J, 2) \cong \mathcal{P}(J)$ is Boolean, whence the Heyting embedding is also Boolean.

The proof will again use the transposed evaluation map,

$$\eta : \mathcal{H} \longrightarrow \mathbb{2}^{\mathsf{DLat}(\mathcal{H},\mathbb{2})} \cong \mathsf{Down}(\mathsf{DLat}(\mathcal{H}, \mathbb{2})^{\mathsf{op}})$$

which, as before, is injective, by the Prime Ideal Theorem (see Lemma 2.6.6). We will use it in the following form due to Birkhoff.

Lemma 2.9.13 (Prime Ideal Theorem)**.** *Let D be a distributive lattice, $I \subseteq D$ an ideal, and $x \in D$ with $x \notin I$. There is a prime ideal $I \subseteq P \subset D$ with $x \notin P$.*

Proof. As in the proof of Lemma 2.6.6, it suffices to prove it for the case $I = (0)$. This time, we use Zorn's Lemma: a poset in which every chain has an upper bound has maximal elements. Consider the poset $\mathcal{I}\backslash x$ of "ideals I without x", $x \notin I$, ordered by inclusion. The union of any chain $I_0 \subseteq I_1 \subseteq ...$ in $\mathcal{I}\backslash x$ is clearly also in $\mathcal{I}\backslash x$, so we have (at least one) maximal element $M \in \mathcal{I}\backslash x$. We claim that $M \subseteq D$ is prime. To that end, take $a, b \in D$ with $a \wedge b \in M$. If $a, b \notin M$, let $M[a] = \{n \leq m \vee a \mid m \in M\}$, the ideal join of M and $\downarrow(a)$, and similarly for $M[b]$. Since M is maximal without x, we therefore have $x \in M[a]$ and $x \in M[b]$. Thus let $x \leq m \vee a$ and $x \leq m' \vee b$ for some $m, m' \in M$. Then $x \vee m' \leq m \vee m' \vee a$ and $x \vee m \leq m \vee m' \vee b$, so taking meets on both sides gives

$$(x \vee m') \wedge (x \vee m) \leq (m \vee m' \vee a) \wedge (m \vee m' \vee b) = (m \vee m') \vee (a \wedge b).$$

Since the righthand side is in the ideal M, so is the left. But then $x \leq x \vee (m \wedge m')$ is also in M, contrary to our assumption that $M \in \mathcal{I}\backslash x$. □

Proof of Theorem 2.9.12. As in (2.41), let $J^{\mathsf{op}} = \mathsf{DLat}(\mathcal{H}, \mathbb{2})$ be the poset of prime filters in $\mathcal{H}$, and consider the transposed evaluation map (2.41),

$$\eta : \mathcal{H} \longrightarrow \mathsf{Down}(\mathsf{DLat}(\mathcal{H}, \mathbb{2})^{\mathsf{op}}) \cong \mathbb{2}^{\mathsf{DLat}(\mathcal{H},\mathbb{2})}$$

given by $\eta(p) = \{F \,|\, p \in F \text{ prime}\} \cong \{f : \mathcal{H} \to \mathbb{2} \,|\, f(p) = 1\}$.

Clearly $\eta(0) = \emptyset$ and $\eta(1) = \mathsf{DLat}(\mathcal{H}, \mathbb{2})$, and similarly for the other meets and joins, so η is a lattice homomorphism. Moreover, if $p \neq q \in \mathcal{H}$ then, as in the proof of 2.6.8, we have that $\eta(p) \neq \eta(q)$, by the Prime Ideal Theorem (Lemma 2.9.13). Thus it only remains to show that

$$\eta(p \Rightarrow q) \;=\; \eta(p) \Rightarrow \eta(q)\,.$$

Unwinding the definitions, this means that, for all $f \in \mathsf{DLat}(\mathcal{H}, \mathbb{2})$,

$$f(p \Rightarrow q) = 1 \quad \text{iff} \quad \text{for all } g \geq f,\ g(p) = 1 \text{ implies } g(q) = 1. \tag{2.42}$$

Equivalently, for all prime filters $F \subseteq \mathcal{H}$,

$$p \Rightarrow q \in F \quad \text{iff} \quad \text{for all prime } G \supseteq F,\ p \in G \text{ implies } q \in G. \tag{2.43}$$

Now if $p \Rightarrow q \in F$, then for all (prime) filters $G \supseteq F$, also $p \Rightarrow q \in G$, and so $p \in G$ implies $q \in G$, since $(p \Rightarrow q) \wedge p \leq q$.

Conversely, suppose $p \Rightarrow q \notin F$, and we seek a prime filter $G \supseteq F$ with $p \in G$ but $q \notin G$. Consider the filter

$$F[p] = \{x \wedge p \leq h \in \mathcal{H} \mid x \in F\}\,,$$

which is the join of F and $\uparrow(p)$ in the poset of filters. If $q \in F[p]$, then $x \wedge p \leq q$ for some $x \in F$, whence $x \leq p \Rightarrow q$, and so $p \Rightarrow q \in F$, contrary to assumption; thus $q \notin F[p]$. By the Prime Ideal Theorem again (applied to the distributive lattice $\mathcal{H}^{\mathsf{op}}$) there is a prime filter $G \supseteq F[p]$ with $q \notin G$. □

Exercise 2.9.14. Give a Kripke countermodel to show that the Law of Excluded Middle $\phi \vee \neg\phi$ is not provable in IPC.

2.10 Frames and locales

Recall that a *supremum* (least upper bound) of $S \subseteq P$ in a poset P is an element $\bigvee S \in P$ such that, for all $y \in S$,

$$\bigvee S \leq y \iff \forall x : S \,.\, x \leq y \;.$$

In particular, $\bigvee \emptyset$ is a least element of P and $\bigvee P$ is a greatest element of P, if they exist.

A poset $(P, \leq)$ is said to be *complete* if it has suprema of *all* subsets. Viewed as a category, P is *both complete and cocomplete* when it is complete as a poset. This is so, first, because coequalizers in a poset always exist, and coproducts are exactly suprema, so a complete poset has all colimits. And moreover, it then also has infima (greatest lower bounds) of arbitrary subsets, and so it is also complete as a category. Indeed, an infimum of $S \subseteq P$ is an element $\bigwedge S \in P$ such that, for all $y \in S$,

$$y \leq \bigwedge S \iff \forall x : S \,.\, y \leq x \;.$$

Proposition 2.10.1. *A poset is complete if, and only if, it has infima $\bigwedge S$ for all subsets $S \subseteq P$.*

Proof. Infima and suprema are expressed in terms of each other as follows:

$$\bigwedge S = \bigvee \left\{x \in P \;\middle|\; \forall y : S \,.\, x \leq y\right\} ,$$
$$\bigvee S = \bigwedge \left\{y \in P \;\middle|\; \forall x : S \,.\, x \leq y\right\} .$$

□

The basic examples of complete posets are the powersets $\mathcal{P}X$, and these are Boolean algebras, and therefore also Heyting. Similarly, the posets of the form $\mathsf{Down}(P)$ of downsets in a poset P are also evidently complete, and we know that these are also Heyting algebras, although not Boolean. This leads us to ask: when is a complete poset P cartesian closed, and therefore a Heyting algebra? Being complete, P has the terminal object, namely the greatest element $\bigvee P = 1 \in P$, and it has binary products which are binary infima. If P is cartesian closed then for all $x, y \in P$ there exists an exponential $(x \Rightarrow y) \in P$, which satisfies, for all $z \in P$,

$$\frac{z \wedge x \leq y}{\overline{z \leq x \Rightarrow y}} \;.$$

First, observe that with the help of this adjunction, we can derive the *infinite distributive law*:

$$x \wedge \textstyle\bigvee_{i \in I} y_i = \bigvee_{i \in I} (x \wedge y_i) \tag{2.44}$$

for any family of elements $\{y_i \in P \mid i \in I\}$, as follows:

$$\frac{\frac{\frac{\frac{x \wedge \bigvee_{i\in I} y_i \leq z}{\bigvee_{i\in I} y_i \leq (x \Rightarrow z)}}{\forall i : I \,.\, (y_i \leq (x \Rightarrow z))}}{\forall i : I \,.\, (x \wedge y_i \leq z)}}{\bigvee_{i\in I}(x \wedge y_i) \leq z}$$

Now since $x \wedge \bigvee_{i\in I} y_i$ and $\bigvee_{i\in I}(x \wedge y_i)$ have the same upper bounds they must be equal (*why?*). Conversely, suppose the distributive law (2.44) holds, and let us *define* $x \Rightarrow y$ to be

$$(x \Rightarrow y) = \bigvee \{z \in P \mid x \wedge z \leq y\} \ . \tag{2.45}$$

Now we can check that this element $x \Rightarrow y$ has the universal property of an exponential (in a poset): for all $x, y \in P$,

$$x \wedge (x \Rightarrow y) \leq y \ , \tag{2.46}$$

and for $z \in P$,

$$x \wedge z \leq y \quad \text{implies} \quad z \leq x \Rightarrow y \ .$$

The implication follows directly from (2.45), and (2.46) follows from the distributive law:

$$x \wedge (x \Rightarrow y) = x \wedge \bigvee \{z \in P \mid x \wedge z \leq y\} = \bigvee \{x \wedge z \mid x \wedge z \leq y\} \leq y \ .$$

Such complete, cartesian closed posets are called *frames.*

Definition 2.10.2. A *frame* is a complete poset satisfying the (infinite) distributive law.

$$x \wedge \bigvee_{i\in I} y_i = \bigvee_{i\in I}(x \wedge y_i) \ .$$

Equivalently, we have just shown that a frame is a complete Heyting algebra. We will distinguish the two, however, by their maps:

A *frame morphism* is a function $f : L \to M$ between frames that preserves finite infima $x \wedge y$ and arbitrary suprema $\bigvee_{i\in I} y_i$. The category of frames and frame morphisms is denoted by Frame.

Warning: just as we need to *require* the preservation of meets $x \wedge y$ although they are determined by the joins $\bigvee_{i\in I} y_i$, a frame morphism need not preserve exponentials $x \Rightarrow y$!

Example 2.10.3. Given a poset P, the downsets $\downarrow P$ form a complete lattice under the inclusion order $S \subseteq T$, and with the set theoretic operations $\bigcup$ and $\bigcap$ as $\bigvee$ and $\bigwedge$. Since $\mathsf{Down}(P)$ is already known to be a Heyting algebra (Example 2.9.4), it is therefore also a frame. (Alternately, we can show that it is a frame by noting that the operations $\bigcup$ and $\cap$ satisfy the infinite distributive law.)

Any monotone map $f : P \to Q$ between posets then gives rise to a frame map

$$\mathsf{Down}(f) : \ \mathsf{Down}(Q) \longrightarrow \mathsf{Down}(P) .$$

(Note the direction!) This is easily seen by recalling that $\mathsf{Down}(P) \cong \mathsf{Pos}(P, \mathbb{2})$ as posets.

Note, moreover, that $\mathsf{Pos}(f, \mathbb{2}) : \mathsf{Pos}(Q, \mathbb{2}) \longrightarrow \mathsf{Pos}(P, \mathbb{2})$ is a (co)limit preserving functor on complete posets, since the (co)limits are just set-theoretic unions and intersections. So $\mathsf{Pos}(f, \mathbb{2})$ therefore has both left and right adjoints. These functors are usually written

$$f_! \dashv f^* \dashv f_* : \mathsf{Pos}(Q, \mathbb{2}) \longrightarrow \mathsf{Pos}(P, \mathbb{2}) .$$

Although it does not in general preserve Heyting implications $S \Rightarrow T$, the monotone map $f^* : \mathsf{Down}(Q) \longrightarrow \mathsf{Down}(P)$ is indeed a morphism of frames. We therefore have a contravariant functor

$$\mathsf{Down}(-) : \ \mathsf{Pos} \to \mathsf{Frame}^{\mathsf{op}}. \tag{2.47}$$

Example 2.10.4. The topology $\mathcal{O}X$ of a topological space X, ordered by inclusion, is a frame, because finite intersections and arbitrary unions of open sets are open. The distributive law holds because intersections distribute over unions. If $f : X \to Y$ is a continuous map between topological spaces, the inverse image map $f^{-1} : \mathcal{O}Y \to \mathcal{O}X$ is a frame homomorphism. Thus, there is a functor

$$\mathcal{O} : \mathsf{Top} \to \mathsf{Frame}^{\mathsf{op}}$$

which maps a space X to its topology $\mathcal{O}X$ and a continuous map $f : X \to Y$ to the inverse image map $f^{-1} : \mathcal{O}Y \to \mathcal{O}X$.

Motivated by the previous examples, we introduce the category of *locales* as the opposite of the category of frames:

$$\mathsf{Loc} = \mathsf{Frame}^{\mathsf{op}} .$$

We can think of a locale as a “generalized space”.

Example 2.10.5. Let P be a poset and define a topology on the elements of P by defining the opens to be the downsets,

$$\mathcal{O}(P) = \mathsf{Down}(P) \cong \mathsf{Pos}(P, \mathbb{2}).$$

These open sets are not only closed under arbitrary unions and finite intersections, but also under *arbitrary* intersections. Such a topological space, in which the opens are closed under all intersections, is said to be an *Alexandroff space* (note that the opens could equivalently be defined to be the upsets). The associated locale is the one in Example 2.10.4, and the associated frame is the one in Example 2.10.3.

Exercise* 2.10.6. This exercise is meant for students with some knowledge of topology. For a topological space X and a point $x \in X$, let $N(x)$ be the neighborhood filter of x,

$$N(x) = \left\{ U \in \mathcal{O}X \;\middle|\; x \in U \right\} .$$

Recall that a T_0-*space* is a topological space X in which points are determined by their neighborhood filters,

$$N(x) = N(y) \Rightarrow x = y \,. \qquad (x, y \in X)$$

Let Top_0 be the full subcategory of Top on T_0-spaces. The functor $\mathcal{O} : \mathsf{Top} \to \mathsf{Loc}$ restricts to a functor $\mathcal{O} : \mathsf{Top}_0 \to \mathsf{Loc}$. Prove that $\mathcal{O} : \mathsf{Top}_0 \to \mathsf{Loc}$ is a faithful functor. Is it full?

Topological semantics for IPC

It should now be clear how to interpret IPC into a topological space X: each formula ϕ is assigned to an open set $[\![\phi]\!] \in \mathcal{O}X$ in such a way that $[\![-]\!]$ is a homomorphism of Heyting algebras.

Definition 2.10.7. A *topological model* of IPC consists of a space X and a function

$$[\![-]\!] : \mathsf{Var} \to \mathcal{O}(X)$$

from the propositional variables $\mathsf{Var} = \{p_0, p_1, \dots\}$ to open sets of X. The interpretation is then extended to all formulas,

$$[\![-]\!] : \mathsf{IPC} \to \mathcal{O}(X)\,,$$

by setting:

$$\begin{aligned}
[\![\top]\!] &= X \\
[\![\bot]\!] &= \emptyset \\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!] \\
[\![\phi \Rightarrow \psi]\!] &= [\![\phi]\!] \Rightarrow [\![\psi]\!]\,.
\end{aligned}$$

The Heyting implication $[\![\phi]\!] \Rightarrow [\![\psi]\!]$ in $\mathcal{O}X$, is defined as in (2.45) as

$$[\![\phi]\!] \Rightarrow [\![\psi]\!] \;=\; \bigcup \bigl\{ U \in \mathcal{O}X \bigm| U \wedge [\![\phi]\!] \leq [\![\psi]\!] \bigr\}\,.$$

Joyal's representation theorem 2.9.12 then easily implies that IPC is sound and complete with respect to topological semantics.

Corollary 2.10.8. *A formula ϕ is provable in* IPC *if, and only if, it holds in every topological interpretation $[\![-]\!]$ into a space X, briefly:*

$$\mathsf{IPC} \vdash \phi \qquad \textit{iff} \qquad [\![\phi]\!] = X \textit{ for all spaces } X\,.$$

Proof. Put the Alexandroff topology on the downsets of prime filters in the Heyting algebra IPC. □

Exercise 2.10.9. Give a topological countermodel to show that the Law of Double Negation $\neg\neg\phi \Rightarrow \phi$ is not provable in IPC.

Modal logic

Chapter 3

First-Order Logic

Having considered equational and propositional logic, we now move on to first-order logic, which is the usual predicate logic with propositional connectives like $\wedge$ and $\Rightarrow$ and the quantifiers $\forall$ and $\exists$. This logic can be seen as propositional logic indexed over an equational theory, in a sense that will become clear; quantification will then be seen to result from completeness with respect to the indexing.

We pursue the same general approach to studying logic via category theory as in the previous chapters, determining categorical structures that model the logical operations, and regarding (certain) categories with these structures as theories, and functors that preserve them as models. We again construct the classifying category for a theory from the syntax of a deductive system and establish its universal property, leading again to functorial semantics. We then establish basic completeness theorems by embedding such classifying categories in particular semantic categories of interest, such as presheaves.

3.1 Predicate logic

Let us first demonstrate our general approach informally with an example. In Chapter **??** we considered models of algebraic theories in categories with finite products. Recall that e.g. a group is a structure of the form:

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

for which, moreover, certain diagrams built from these basic arrows must commute. We can express some properties of groups in terms of further equations, for example commutativity

$$x \cdot y = y \cdot x\ ,$$

which is expressed by the diagram

$$\begin{tikzcd}
G \times G \arrow[r, "\sim"] \arrow[d, "m"'] & G \times G \arrow[d, "m"] \\
G \arrow[r, equal] & G
\end{tikzcd}$$

where the iso on top is the familiar "twist" map permuting the factors.

As we saw, such equations can be interpreted in any category with finite products, providing a large scope for categorical semantics of algebraic theories. However, there are also many significant properties of algebraic structures which cannot be expressed merely with equations. Consider the statement that a group (G, m, i, e) has no non-trivial square roots of unity,

$$\forall\, x : G \,.\, (x \cdot x = e \;\Rightarrow\; x = e) \;. \tag{3.1}$$

This is a simple first-order logical statement which cannot be rewritten as a system of equations (how would one prove that?). To see what its categorical interpretation ought to be, let us look at its usual set-theoretic interpretation. Each of subformulas $x \cdot x = e$ and $x = e$ determines a subset of G,

$$\begin{tikzcd}
\bigl\{x \in G \bigm| x \cdot x = e\bigr\} \arrow[r, hook, dotted] \arrow[d, hook, "i"'] & \bigl\{x \in G \bigm| x = e\bigr\} \arrow[d, hook, "j"'] \\
G \arrow[r, equal] & G.
\end{tikzcd}$$

The implication $x \cdot x = e \Rightarrow x = e$ holds just when $\bigl\{x \in G \bigm| x \cdot x = e\bigr\}$ is contained in $\bigl\{x \in G \bigm| x = e\bigr\}$. In categorical language, the inclusion i *factors through* the inclusion j. Observe that such a factorization is unique, if it exists. The defining formulas of the subsets $\bigl\{x \in G \bigm| x \cdot x = e\bigr\}$ and $\bigl\{x \in G \bigm| x = e\bigr\}$ are equations, and so the subsets themselves can be constructed as equalizers (interpreting $\cdot$ as m as above):

$$\begin{tikzcd}
\bigl\{x \in G \bigm| x \cdot x = e\bigr\} \arrow[r, hook] & G \arrow[r, "\langle 1_G, 1_G \rangle"] \arrow[rr, bend right=20, "e\,!_G"'] & G \times G \arrow[r, "m"] & G
\end{tikzcd}$$

$$\begin{tikzcd}
\bigl\{x \in G \bigm| x = e\bigr\} \arrow[r, hook] & G \arrow[r, shift left, "1_G"] \arrow[r, shift right, "e\,!_G"'] & G
\end{tikzcd}$$

In sum, we can interpret condition (3.1) in any category with products and equalizers, i.e. in any category with all finite limits.[1] This allows us to define the notion of a group without square roots of unity in any category $\mathcal{C}$ with finite limits as an object G with morphisms $m : G \times G \to G$ and $i : G \to G$ and $e : 1 \to G$, such that (G, m, i, e) is a group in $\mathcal{C}$, *and* the equalizer of $m \circ \langle 1_G, 1_G \rangle$ and $e\,!_G$ factors through that of 1_G and $e\,!_G$.

The aim of this chapter is to analyze how such examples can be treated systematically. We will relate (various fragments of) first-order logic to categorical structures that are suitable for the interpretation of the logic. The general outline will be as follows:

[1] We are *not* saying that finite limits suffice to interpret arbitrary formulas built from universal quantifiers and implications. The formula at hand has the special form $\forall\, x \,.\, (\varphi(x) \Longrightarrow \psi(x))$, where $\varphi(x)$ and $\psi(x)$ do not contain any further $\forall$ or $\Longrightarrow$.

1. A language $\mathcal{L}$ for a first-order theory consists, as usual, of some basic relation, function, and constant symbols, say $\mathcal{L} = (R, f, c)$.

2. An $\mathcal{L}$-structure in a category $\mathcal{C}$ with finite limits is an interpretation of $\mathcal{L}$ in $\mathcal{C}$ as an object M equipped with corresponding relations (subobjects) and operations (morphisms) of appropriate arities,

$$\begin{aligned} &R^M \rightarrowtail M \times \cdots \times M \\ &f^M : M \times \cdots \times M \longrightarrow M \\ &c^M : 1 \to M. \end{aligned}$$

3. Formulas φ in first-order logic will be interpreted as subobjects,

$$[\![\varphi]\!] \rightarrowtail M \times \cdots \times M.$$

   The interpretation makes use of categorical operations in $\mathcal{C}$ corresponding to the logical ones appearing in the formula φ.

4. A theory $\mathbb{T}$ in first-order logic consists of a set of (binary) sequents,

$$\varphi \vdash \psi.$$

5. A model of $\mathbb{T}$ is then an interpretation M in which the corresponding subobjects "satisfy" all the sequents of $\mathbb{T}$, in the sense that

$$[\![\varphi]\!] \leq [\![\psi]\!] \qquad \text{in } \mathsf{Sub}(M^n).$$

6. We shall give a deductive calculus for such sequents $\varphi \vdash \psi$, prove that it is sound with respect to categorical models, and then use it to construct a classifying category $\mathcal{C}_{\mathbb{T}}$ with the expected universal property: models of $\mathbb{T}$ in any suitably-structured category $\mathcal{C}$ correspond uniquely to structure-preserving functors $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$.

7. Completeness of the calculus with respect to general models follows from classification, while completeness with respect to special models, such as "Kripke-models" Set^K, follows from embedding $\mathcal{C}_{\mathbb{T}}$ in such special categories.

Not only does having such categorical semantics permit us to prove things about different systems of logic (such as consistency of formal systems and independence of axioms), it also allows us to *use* the systems of logic to reason formally about logical structures in categories of various kinds.

3.1.1 Theories

A *first-order theory* $\mathbb{T}$ consists of an underlying *type theory* and a set of formulas in a *fragment of first-order logic*. Anticipating Chapter 5, the type theory is given by a set of basic types, a set of basic constants together with their types, rules for forming types, and rules and axioms for deriving typing judgments,

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

expressing that term t has type B in typing context $x_1 : A_1$, …, $x_n : A_n$. There is also a set of axioms and rules of inference which tell us which equations between terms,

$$x_1 : A_1, \ldots, x_n : A_n \mid t = u : B \; ,$$

are assumed to hold. This part of the theory $\mathbb{T}$ may be regarded as providing the underlying structure, on top of which the logical formulas are defined. For first-order logic, the underlying type theory is essentially the same as the equational logic that we already met in Chapter **??**.

A fragment of first-order logic is then given by a set of *basic relation symbols*, together with a specification of which first-order operations are to be used in building formulas. Each basic relation symbol has a *signature* $(A_1, \ldots, A_n)$, which specifies the types of its arguments. The *arity* of a relation symbol is the number of arguments it takes. The judgment[2]

$$x_1 : A_1, \ldots, x_n : A_n \mid \phi \; \mathtt{pred}$$

states that ϕ is a well-formed formula in typing context $x_1 : A_1, \ldots, x_n : A_n$. For each basic relation symbol R with signature $(A_1, \ldots, A_n)$ there is an inference rule

$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid R(t_1, \ldots, t_n) \; \mathtt{pred}}$$

which says that the atomic formula $R(t_1, \ldots, t_n)$ is well formed in context Γ. Depending on what fragment of first-order logic is involved, there may be other rules for forming logical formulas. For example, if equality is present as a formula, then for each type A there is a rule:

$$\frac{\Gamma \mid t : A \qquad \Gamma \mid u : A}{\Gamma \mid t =_A u \; \mathtt{pred}}$$

And if conjunction is present, then there is a rule:

$$\frac{\Gamma \mid \varphi \; \mathtt{pred} \qquad \Gamma \mid \psi \; \mathtt{pred}}{\Gamma \mid \varphi \wedge \psi \; \mathtt{pred}}$$

Other such rules will be given when we come to the study of particular logical operations.

[2] We follow type-theoretic practice here by adding the tag `pred` to turn what would otherwise be an exhibited formula in context into a judgement concerning the formula.

The basic logical judgment of a first-order theory is *entailment* between formulas,

$$x_1 : A_1, \ldots, x_n : A_n \mid \varphi_1, \ldots, \varphi_m \vdash \psi \ ,$$

which states that in the typing context $x_1 : A_1, \ldots, x_n : A_n$, the assumptions φ_1, …, φ_m entail ψ. It is understood that the terms appearing in the formulas are well-typed in the typing context, and that the formulas $\varphi_1, \ldots, \varphi_m, \psi$ are part of the fragment of the logic of $\mathbb{T}$. When the fragment contains conjunction $\wedge$ it is convenient to restrict attention to *binary* sequents,

$$x_1 : A_1, \ldots, x_n : A_n \mid \varphi \vdash \psi,$$

by replacing $\varphi_1, \ldots, \varphi_m$ with $\varphi_1 \wedge \ldots \wedge \varphi_m$. When the fragment contains equality, we may replace the type-theoretic equality judgments

$$x_1 : A_1, \ldots, x_n : A_n \mid t = u : B$$

with the entailments

$$x_1 : A_1, \ldots, x_n : A_n \mid \cdot \vdash t =_B u \ .$$

The subscript at the equality sign indicates the type at which the equality is taken. In a theory $\mathbb{T}$ there are basic entailments, or axioms, which together with the inference rules for the opertations involved can be used for deriving judgments, as usual.

We shall consider several standard fragments of first-order logic, determined by selecting a subset of logical connectives and quantifiers. These are as follows:

1. *Full first-order logic* consists of formulas built from the logical operations

$$= \quad \top \quad \bot \quad \neg \quad \wedge \quad \vee \quad \Rightarrow \quad \forall \quad \exists \, .$$

2. *Cartesian logic* is the fragment

$$= \quad \top \quad \wedge \, .$$

3. *Regular logic* is the fragment

$$= \quad \top \quad \wedge \quad \exists \, .$$

4. *Coherent logic* is the fragment built from

$$= \quad \top \quad \wedge \quad \exists \quad \bot \quad \vee \, .$$

5. *Geometric logic* consists of formulas of the form

$$\forall x : A \, . \, (\varphi \Rightarrow \psi) \ ,$$

where φ and ψ are coherent formulas.[3]

[3] There is also *infinitary geometric logic*, in which φ and ψ may contain disjunctions $\bigvee_i \vartheta_i$ of infinitely many formulas ϑ_i.

The names for these fragments come from the names of the various kinds of categories in which they are interpreted. We shall also consider both *Heyting* and *Boolean* theories in full first-order logic, which differ according to their assumed rules of inference and their intended interpretations.

The well-formed terms and formulas of a first-order theory $\mathbb{T}$ constitute its *language*. It may seem that we are doing things backwards, because we should have spoken of first-order languages before we spoke of first-order theories. While this is possible for simple theories, it becomes difficult to do when we consider more complicated theories in which types and logical formulas are intertwined. In such cases the typing judgments and entailments may be given by a mutual recursive definition. In order to find out whether a given term is well-formed, we might have to prove a logical statement. In everyday mathematics this occurs all the time, for example, to show that the term $\int_0^\infty f$ denotes a real number, it may be necessary to prove that $f : \mathbb{R} \to \mathbb{R}$ is an integrable function and that the integral has a finite value. This is why it does not always make sense to strictly differentiate a language from a theory.[4]

In order to focus on the logical part of first-order theories, we will limit attention to only two very simple kinds of type theory. A *single-sorted* first-order theory has as its underlying type theory a single type A, and for each $k \in \mathbb{N}$ a set of basic k-ary function symbols. The rules for typing judgments are:

1. Variables in contexts:

$$\frac{}{x_1 : A, \dots, x_n : A \mid x_i : A}$$

2. For each basic function symbol f of arity k, there is an inference rule

$$\frac{\Gamma \mid t_1 : A \qquad \cdots \qquad \Gamma \mid t_n : A}{\Gamma \mid f(t_1, \dots, t_n) : A}$$

This much is essentially an algebraic theory. In addition, however, a single-sorted first-order theory may contain relation symbols, formulas, axioms, and rules of inference which an algebraic theory does not.

A slight generalization of a single-sorted theory is a *many-sorted* one. Its underlying type theory is given by a set of types, and a set of basic function symbols. Each function symbol f has a *signature* $(A_1, \dots, A_n; B)$, where n is the arity of f and $A_1, \dots, A_n, B$ are types. The rules for typing judgments are:

1. Variables in contexts:

$$\frac{}{x_1 : A_1, \dots, x_n : A_n \mid x_i : A_i}$$

2. For each basic function symbol f with signature $(A_1, \dots, A_n; B)$, there is an inference rule

$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid f(t_1, \dots, t_n) : B}$$

[4] However, it *does* make sense to distinguish *syntax* from theories. Rules of substitution and the behaviour of free and bound variables are syntactic considerations, for example.

We may write suggestively $f : A_1 \times \cdots \times A_n \to B$ to indicate that $(A_1, \ldots, A_n; B)$ is the signature of f. However, this does not mean that $A_1 \times \cdots \times A_n \to B$ is a type! A many-sorted first-order theory does *not* have any type forming operations, such as × and →. We shall consider type theories with such operations in Chapter 5.

3.1.2 Subobjects

Formulas of first-order logic will be interpreted as "generalized subsets", i.e. subobjects. We therefore need to review some of the basic theory of these.

Let A be an object in a category $\mathcal{C}$. If $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ are monos into A, we say that i is smaller than j, and write $i \le j$, when there exists a morphism $k : I \to J$ such that the following diagram commutes:

$$\begin{tikzcd}
I \arrow[rr, dotted, "k"] \arrow[dr, tail, "i"'] & & J \arrow[dl, tail, "j"] \\
& A &
\end{tikzcd}$$

If such a k exists then it, too, is monic, since i is, and it is unique, since j is monic. The class $\mathsf{Mono}(A)$ of all monos into A is thus preordered by the relation $\le$. It is the same as the slice category $\mathsf{Mono}(\mathcal{C})/A$ consisting of all monos with codomain A and commutative triangles between them. Let $\mathsf{Sub}(A)$ be the poset reflection of the preorder $\mathsf{Mono}(A)$. Thus the elements of $\mathsf{Sub}(A)$ are equivalence classes of monos into A, where $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ are equivalent when $i \le j$ and $j \le i$ (note that then $I \cong J$). The induced relation $\le$ on $\mathsf{Sub}(A)$ is then a partial order.

We have to be a bit careful with the formation of $\mathsf{Sub}(A)$, since it is defined as a quotient of a *class* $\mathsf{Mono}(A)$. In many particular cases the general construction by quotients can be avoided. If we can demonstrate that the preorder $\mathsf{Mono}(A)$ is equivalent, as a category, to a poset P then we can simply take $\mathsf{Sub}(A) = P$. We will usually simply require that $\mathsf{Sub}(A)$ is small.

Definition 3.1.1. A category $\mathcal{C}$ is *well-powered* when, for all $A \in \mathcal{C}$, there is at most a *set* of subobjects of A, so that the category $\mathsf{Mono}(A)$ is equivalent to a (small) poset. In other words, $\mathsf{Sub}(A)$ is a small category for every $A \in \mathcal{C}$.

We shall often speak of subobjects as if they were monos rather than equivalence classes of monos. It is then understood that we mean the subobjects represented by monos and not the monos themselves. Sometimes we refer to a mono $i : I \rightarrowtail A$ by its domain I only, even though the object I itself does not determine the morphism i. Hopefully this will not cause confusion, as it is always going to be clear which mono is meant to go along with the object I.

In a category $\mathcal{C}$ with finite limits the assignment $A \mapsto \mathsf{Sub}(A)$ is the object part of the contravariant *subobject functor*,

$$\mathsf{Sub} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Poset}\ .$$

The morphism part of Sub is given by pullback; in detail, given any $f : A \to B$, let $\mathsf{Sub}(f) = f^* : \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ be the monotone map that takes the subobject (represented by) $i : I \rightarrowtail B$ to the subobject (represented by) $f^*i : f^*I \rightarrowtail A$, where $f^*i : f^*I \rightarrowtail A$ is a pullback of i along f:

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

Recall that a pullback of a mono is again mono, so this definition makes sense. We need to verify (1) that if two monos $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ are equivalent, then their pullbacks are so as well; and (2) that $\mathsf{Sub}(1_A) = 1_{\mathsf{Sub}(A)}$ and $\mathsf{Sub}(g \circ f) = \mathsf{Sub}(f) \circ \mathsf{Sub}(g)$. These all follow easily from the fact that pullback is a functor $\mathcal{C}/B \to \mathcal{C}/A$, which reduces to the familiar "2-pullbacks" lemma:

Lemma 3.1.2. *Suppose both squares in the following diagram are pullbacks:*

$$\begin{tikzcd}
\cdot \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \cdot \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \cdot \arrow[d] \\
\cdot \arrow[r] & \cdot \arrow[r] & \cdot
\end{tikzcd}$$

Then the outer rectangle is a pullback diagram as well. Moreover, if the outer rectangle and the right square are pullbacks, then so is the left square.

Proof. This is left as an exercise in diagram chasing. □

Of course, pullbacks are really only determined up to isomorphism, but this does not cause any problems because isomorphic monos represent the same subobject.

In the semantics to be given below, a formula

$$x : A \mid \varphi \ \texttt{pred}$$

will be interpreted as a subobject

$$[\![x : A \mid \varphi]\!] \rightarrowtail [\![A]\!].$$

Thus $\mathsf{Sub}(A)$ can be regarded as the poset of "predicates" on A, generalizing the powerset of a set A. Logical operations on formulas then correspond to operations on $\mathsf{Sub}(A)$. The structure of $\mathsf{Sub}(A)$ therefore determines which logical connectives can be interpreted. If $\mathsf{Sub}(A)$ is a Heyting algebra, then we can interpret the (propositional part of) the full intuitionistic propositional calculus (cf. Subsection 2.9), but if $\mathsf{Sub}(A)$ only has binary meets, then all that can be interpreted are $\top$ and $\wedge$. We will work out details of different operations in the following sections, but one common aspect that we require is the "stability" of the interpretation of the logical operations, in a sense that we now make clear.

Substitution and stability

Let us consider the interpretation of substitution of terms for variables. There are two kinds of substitution, into a term, and into a formula. We may substitute a term $x : A \mid t : B$ for a variable y in a term $y : B \mid u : C$ to obtain a new term $x : A \mid u[t/y] : C$. If t and u are interpreted as morphisms

$$\begin{tikzcd}
\llbracket A \rrbracket \arrow[r, "\llbracket t \rrbracket"] & \llbracket B \rrbracket \arrow[r, "\llbracket u \rrbracket"] & \llbracket C \rrbracket
\end{tikzcd}$$

then $u[t/y]$ is interpreted as their composition:

$$\llbracket x : A \mid u[t/y] : C \rrbracket = \llbracket y : B \mid u : C \rrbracket \circ \llbracket x : A \mid t : B \rrbracket \ .$$

Thus, *substitution into a term is composition.*

The second kind of substitution occurs when we substitute a term $x : A \mid t : B$ for a variable y in a formula $y : B \mid \varphi$ to obtain a new formula $x : A \mid \varphi[t/y]$. If t is interpreted as a morphism $\llbracket t \rrbracket : \llbracket A \rrbracket \to \llbracket B \rrbracket$ and φ is interpreted as a subobject $\llbracket \varphi \rrbracket \rightarrowtail \llbracket B \rrbracket$ then the interpretation of $\varphi[t/y]$ is the pullback of $\llbracket \varphi \rrbracket$ along $\llbracket t \rrbracket$:

$$\begin{tikzcd}
\llbracket \varphi[t/y] \rrbracket = \llbracket t \rrbracket^* \llbracket \varphi \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \llbracket \varphi \rrbracket \arrow[d, tail] \\
\llbracket A \rrbracket \arrow[r, "\llbracket t \rrbracket"'] & \llbracket B \rrbracket
\end{tikzcd}$$

Thus, *substitution into a formula is pullback*,

$$\llbracket x : A \mid \varphi[t/y] \, \rrbracket = \llbracket x : A \mid t : B \rrbracket^* \llbracket y : B \mid \varphi \rrbracket.$$

Now, because substitution respects the syntactical, logical operations, e.g.

$$(\varphi \wedge \psi)[t/x] = \varphi[t/x] \wedge \psi[t/x],$$

the categorical structures used to interpret the various logical operations such as $\wedge$ must also behave well with respect to the interpretation of substitution, i.e. pullback. We say that a categorical property or structure is *stable (under pullbacks)* if it is preserved by pullbacks, so that e.g.

$$\begin{aligned}
\llbracket t \rrbracket^* \llbracket (\varphi \wedge \psi) \rrbracket = \llbracket (\varphi \wedge \psi)[t/x] \rrbracket &= \llbracket \varphi[t/x] \wedge \psi[t/x] \rrbracket \\
&= \llbracket \varphi[t/x] \rrbracket \wedge \llbracket \psi[t/x] \rrbracket = \llbracket t \rrbracket^* \llbracket \varphi \rrbracket \wedge \llbracket t \rrbracket^* \llbracket \psi \rrbracket \ .
\end{aligned}$$

In more detail, say that a category $\mathcal{C}$ has *stable meets* if each poset $\mathsf{Sub}(A)$ has binary meets, and the pullback of a meet $I \wedge J \rightarrowtail A$ along any map $f : B \to A$ is the meet $f^*I \wedge f^*J \rightarrowtail A$ of the respective pullbacks,

$$f^*(I \wedge J) = f^*I \wedge f^*J.$$

This means that the meet operation,

$$\wedge : \mathsf{Sub}(A) \times \mathsf{Sub}(A) \longrightarrow \mathsf{Sub}(A)$$

is natural in A, in the sense that for any map $f : B \to A$ the following diagram commutes.

$$\begin{tikzcd}
\mathsf{Sub}(A) \times \mathsf{Sub}(A) \arrow[r, "\wedge_A"] \arrow[d, "f^* \times f^*"'] & \mathsf{Sub}(A) \arrow[d, "f^*"] \\
\mathsf{Sub}(B) \times \mathsf{Sub}(B) \arrow[r, "\wedge_B"'] & \mathsf{Sub}(B)
\end{tikzcd}$$

Exercise 3.1.3. Show that any category $\mathcal{C}$ with finite limits has stable meets in the foregoing sense: each poset $\mathsf{Sub}(A)$ has all finite meets (i.e. including the "empty meet" $\mathbf{1}$), and these are stable under pullbacks. Conclude that for any finite limit category $\mathcal{C}$, the subobject functor $\mathsf{Sub} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Pos}$ therefore factors through the subcategory of $\wedge$-semilattices.

Generalized elements

In any category, we can regard arbitrary arrows $x : X \to C$ as *generalized elements* of C, thinking thereby of variable elements or parameters. With respect to a subobject $S \rightarrowtail C$, such an element is said to be *in the subobject*, writttten

$$x \in_C S,$$

if it factors through (any mono representing) the subobject,

$$\begin{tikzcd}
 & S \arrow[d, tail] \\
X \arrow[ur, dotted] \arrow[r, "x"'] & C
\end{tikzcd}$$

which, observe, it then does uniquely. The following "generalized element semantics" can be a useful technique for "externalizing" the operations on subobjects into statements about generalized elements.

Proposition 3.1.4. *Let C be any object in a category $\mathcal{C}$ with finite limits.*

1. *for the top element $\mathbf{1} \in \mathsf{Sub}(C)$, and for all $x : X \to C$,*

$$x \in_C \mathbf{1}.$$

2. *for any $S, T \in \mathsf{Sub}(C)$,*

$$S \leq T \iff x \in_C S \text{ implies } x \in_C T, \text{ for all } x : X \to C.$$

3. *for any* $S, T \in \mathsf{Sub}(C)$*, and for all* $x : X \to C$*,*

$$x \in_C S \wedge T \iff x \in_C S \text{ and } x \in_C T.$$

4. *for the subobject* $\Delta = [\langle 1_C, 1_C\rangle] \in \mathsf{Sub}(C \times C)$*, and for all* $x, y : X \to C$*,*

$$\langle x, y\rangle \in \Delta \iff x = y.$$

5. *for the equalizer* $E_{(f,g)} \rightarrowtail A$ *of a pair of arrows* $f, g : A \rightrightarrows B$*, and for all* $x : X \to A$*,*

$$x \in_A E_{(f,g)} \iff fx = gx.$$

6. *for the pullback* $f^*S \rightarrowtail A$ *of a subobject* $S \rightarrowtail B$ *along any arrow* $f : A \to B$*, and for all* $x : X \to A$*,*

$$x \in_A f^*S \iff fx \in_B S.$$

Exercise 3.1.5. Prove the proposition.

3.1.3 Cartesian logic

We begin with a basic system of logic for categories with finite limits, also called *cartesian categories*, which we therefore call *cartesian logic*. This is a logic of formulas built from the logical operations $=$, $\top$, and $\wedge$, over a multi-sorted type theory with unit type $\mathtt{1}$. (See section **??** for multi-sorted type theories and the axioms for the unit type. In a dependently-typed formulation as in Chapter **??** one would also include equality types.).

Formation rules for cartesian logic

Given a basic language consisting of a stock of relation and function symbols (with arities), the terms are built up as explained in Section 3.1.1 from the basic function symbols and variables (we take "constants" to be 0-ary function symbols). The rules for constructing formulas are as follows:

1. The 0-ary relation symbol $\top$ is a formula:

$$\frac{}{\Gamma \mid \top \ \mathtt{pred}}$$

2. For each basic relation symbol R with signature $(A_1, \ldots, A_n)$ there is a rule

$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid R(t_1, \ldots, t_n) \ \mathtt{pred}}$$

3. For each type A, there is a rule

$$\frac{\Gamma \mid s : A \qquad \Gamma \mid t : A}{\Gamma \mid s =_A t \ \mathtt{pred}}$$

4. Conjunction:
$$\frac{\Gamma \mid \varphi \ \texttt{pred} \qquad \Gamma \mid \psi \ \texttt{pred}}{\Gamma \mid \varphi \wedge \psi \ \texttt{pred}}$$

5. Weakening:
$$\frac{\Gamma \mid \varphi \ \texttt{pred}}{\Gamma, x : A \mid \varphi \ \texttt{pred}}$$

Observe that, as usual, there is then a derived operation of substitution of terms for variables into formulas, defined by structural recursion on the above specification of formulas:

Substitution:
$$\frac{\Gamma \mid t : A \qquad \Gamma, x : A \mid \varphi \ \texttt{pred}}{\Gamma \mid \varphi[t/x] \ \texttt{pred}}$$

Inference rules for cartesian logic

Although we do not yet need them, we state the rules of inference here, too, for the convenience of having the entire specification of cartesian logic in one place. As already mentioned, we can conveniently state this deductive calculus using only *binary* sequents,

$$\Gamma \mid \psi \vdash \varphi.$$

We omit mention of the context Γ when it is the same in the premisses and conclusion of a rule.

1. Weakening:
$$\frac{\Gamma \mid \psi \vdash \varphi}{\Gamma, x : A \mid \psi \vdash \varphi}$$

2. Substitution:
$$\frac{\Gamma \mid t : A \qquad \Gamma, x : A \mid \psi \vdash \varphi}{\Gamma \mid \psi[t/x] \vdash \varphi[t/x]}$$

3. Identity:
$$\frac{}{\varphi \vdash \varphi}$$

4. Cut:
$$\frac{\psi \vdash \theta \qquad \theta \vdash \varphi}{\psi \vdash \varphi}$$

5. Equality:
$$\frac{}{\psi \vdash t =_A t} \qquad \frac{\psi \vdash t =_A u \qquad \psi \vdash \varphi[t/z]}{\psi \vdash \varphi[u/z]}$$

6. Truth:
$$\frac{}{\psi \vdash \top}$$

7. Conjunction:

$$\frac{\vartheta \vdash \varphi \qquad \vartheta \vdash \psi}{\vartheta \vdash \varphi \wedge \psi} \qquad \frac{\vartheta \vdash \varphi \wedge \psi}{\vartheta \vdash \psi} \qquad \frac{\vartheta \vdash \varphi \wedge \psi}{\vartheta \vdash \varphi}$$

Exercise 3.1.6. Derive symmetry and transitivity of equality:

$$\frac{\Gamma \mid \psi \vdash t =_A u}{\Gamma \mid \psi \vdash u =_A t} \qquad \frac{\Gamma \mid \psi \vdash t =_A u \qquad \Gamma \mid \psi \vdash u =_A v}{\Gamma \mid \psi \vdash t =_A v}$$

Example 3.1.7. The theory of a poset is a cartesian theory. There is one basic sort $\mathtt{P}$ and one binary relation symbol $\leq$ with signature $(\mathtt{P}, \mathtt{P})$. The axioms are the familiar axioms for reflexivity, transitivity, and antisymmetry:

$$x : \mathtt{P} \mid \cdot \vdash x \leq x$$
$$x : \mathtt{P}, y : \mathtt{P}, z : \mathtt{P} \mid x \leq y \wedge y \leq z \vdash x \leq z$$
$$x : \mathtt{P}, y : \mathtt{P} \mid x \leq y \wedge y \leq x \vdash x =_{\mathtt{P}} y$$

There are also many examples, such as ordered groups, ordered fields, etc., that extend the theory of posets with further algebraic operations and equations.

Example 3.1.8. An *equivalence relation* in a cartesian category is a model of the corresponding theory with one basic sort $\mathtt{A}$ and one binary relation symbol $\sim$ with signature $(\mathtt{A}, \mathtt{A})$. The axioms are the familiar axioms for reflexivity, symmetry, and transitivity:

$$x : \mathtt{A} \mid \cdot \vdash x \sim x$$
$$x : \mathtt{A}, y : \mathtt{A} \mid x \sim y \vdash y \sim x$$
$$x : \mathtt{A}, y : \mathtt{A}, z : \mathtt{A} \mid x \sim y \ \wedge \ y \sim z \vdash x \sim z$$

Semantics of cartesian logic

In order to give the semantics of cartesian logic, we require a couple of useful propositions regarding cartesian categories.

Proposition 3.1.9. *If a category $\mathcal{C}$ has pullbacks then, for every $A \in \mathcal{C}$, the poset $\mathsf{Sub}(A)$ has finite limits. Moreover, these are stable under pullback.*

Proof. The poset $\mathsf{Sub}(A)$ has finite limits if it has a top object and binary meets. The top object of $\mathsf{Sub}(A)$ is the subobject $[1_A : A \to A]$. The meet of subobjects $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ is the subobject $i \wedge j = i \circ (i^* j) = j \circ (j^* i) : I \wedge J \rightarrowtail A$ obtained by pullback, as in the following diagram:

$$\begin{tikzcd}
I \wedge J \arrow[r, tail, "j^* i"] \arrow[d, tail, "i^* j"'] \arrow[dr, phantom, "\lrcorner", very near start] & J \arrow[d, tail, "j"] \\
I \arrow[r, tail, "i"'] & A
\end{tikzcd}$$

It is easy to verify that $I \wedge J$ is the infimum of I and J. Finally, stability follows from a familiar diagram chase on a cube of pullbacks. □

Proposition 3.1.10. *A category has has all finite limits just if it has all finite products and pullbacks of monos along monos.*

Proof. It is sufficient to show that the category has equalizers. To construct the equalizer of parallel arrows $f : A \to B$ and $g : A \to B$, first observe that the arrows

$$A \xrightarrow{\langle 1_A, f\rangle} A \times B \qquad\qquad A \xrightarrow{\langle 1_A, g\rangle} A \times B$$

are monos because the projection $\pi_0 : A \times B \to A$ is their left inverse. Therefore, we may construct the pullback

$$\begin{tikzcd}
P \arrow[r, tail, "p"] \arrow[d, tail, "q"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, tail, "\langle 1_A, f\rangle"] \\
A \arrow[r, tail, "\langle 1_A, g\rangle"'] & A \times B
\end{tikzcd}$$

The morphisms p and q coincide because $\langle 1_A, f\rangle$ and $\langle 1_A, g\rangle$ have a common left inverse π_0:

$$p = 1_A \circ p = \pi_0 \circ \langle 1_A, f\rangle \circ p = \pi_0 \circ \langle 1_A, f\rangle \circ q = 1_A \circ q = q\ .$$

Let us show that $p : P \to A$ is the equalizer of f and g. First, p equalizes f and g,

$$f \circ p = \pi_1 \circ \langle 1_A, f\rangle \circ p = \pi_1 \circ \langle 1_A, g\rangle \circ q = g \circ q = g \circ p\ .$$

If $k : K \to A$ also equalizes f and g then

$$\langle 1_A, f\rangle \circ k = \langle k, f \circ k\rangle = \langle k, g \circ k\rangle = \langle 1_A, g\rangle \circ k\ ,$$

therefore by the universal property of the constructed pullback there exists a unique factorization $\overline{k} : K \to P$ such that $k = p \circ \overline{k}$, as required. □

We now explain how cartesian logic is interpreted in a cartesian category $\mathcal{C}$ (i.e. $\mathcal{C}$ is finitely complete). Let $\mathbb{T}$ be a multi-sorted cartesian theory. Recall that the type theory of $\mathbb{T}$ is specified by a set of sorts (types) $A, ...$ and a set of basic function symbols $f, ...$ together with their signatures, while the logic is given by a set of basic relation symbols $R, ...$ with their signatures, and a set of axioms in the form of logical entailments between formulas in context,

$$\Gamma \mid \psi \vdash \varphi.$$

Definition 3.1.11. An *interpretation* of $\mathbb{T}$ in $\mathcal{C}$ is given by the following data, where Γ stands for a typing context $x_1 : A_1, \ldots, x_n : A_n$, and ψ and φ are formulas:

1. Each sort A is interpreted as an object $[\![A]\!]$, with the unit sort $\mathtt{1}$ being interpreted as the terminal object $\mathtt{1}$.

2. A typing context $x_1 : A_1, \ldots, x_n : A_n$ is interpreted as the product $\llbracket A_1 \rrbracket \times \cdots \times \llbracket A_n \rrbracket$. The empty context is interpreted as the terminal object $\mathsf{1}$.

3. A basic function symbol f with signature $(A_1, \ldots, A_m; B)$ is interpreted as a morphism $\llbracket f \rrbracket : \llbracket A_1 \rrbracket \times \cdots \llbracket A_m \rrbracket \to \llbracket B \rrbracket$.

4. A basic relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted as a subobject $\llbracket R \rrbracket \in \mathsf{Sub}(\llbracket A_1 \rrbracket \times \cdots \times \llbracket A_n \rrbracket)$.

We then extend the interpretation to all terms and formulas as follows:

1. A term in context $\Gamma \mid t : B$ is interpreted as a morphism

$$\llbracket \Gamma \mid t : B \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket B \rrbracket$$

according to the following specification.

- A variable $x_0 : A_1, \ldots, x_n : A_n \mid x_i : A_i$ is interpreted as the i-th projection $\pi_i : \llbracket A_1 \rrbracket \times \cdots \times \llbracket A_n \rrbracket \to \llbracket A_i \rrbracket$.
- The interpretation of $\Gamma \mid * : \mathsf{1}$ is the unique morphism $!_{\llbracket \Gamma \rrbracket} : \llbracket \Gamma \rrbracket \to \mathsf{1}$.
- A composite term $\Gamma \mid f(t_1, \ldots, t_m) : B$, where f is a basic function symbol with signature $(A_1, \ldots, A_m; B)$, is interpreted as the composition

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, "{\langle \llbracket t_1 \rrbracket, \ldots, \llbracket t_m \rrbracket \rangle}"] & \llbracket A_1 \rrbracket \times \cdots \times \llbracket A_m \rrbracket \arrow[r, "\llbracket f \rrbracket"] & \llbracket B \rrbracket
\end{tikzcd}$$

Here $\llbracket t_i \rrbracket$ is shorthand for $\llbracket \Gamma \mid t_i : A_i \rrbracket$.

2. A formula in a context $\Gamma \mid \varphi$ is interpreted as a subobject $\llbracket \Gamma \mid \varphi \rrbracket \in \mathsf{Sub}(\llbracket \Gamma \rrbracket)$ according to the following specification.

- The logical constant $\top$ is interpreted as the maximal subobject, represented by the identity arrow:
$$\llbracket \Gamma \mid \top \rrbracket = [\, \mathsf{1}_{\llbracket \Gamma \rrbracket} : \llbracket \Gamma \rrbracket \to \llbracket \Gamma \rrbracket \,]$$
- An atomic formula $\Gamma \mid R(t_1, \ldots, t_m)$, where R is a basic relation symbol with signature $(A_1, \ldots, A_m)$ is interpreted as the left vertical arrow in the following pullback square:

$$\begin{tikzcd}
\llbracket \Gamma \mid R(t_1, \ldots, t_m) \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \llbracket R \rrbracket \arrow[d, tail] \\
\llbracket \Gamma \rrbracket \arrow[r, "{\langle \llbracket t_1 \rrbracket, \ldots, \llbracket t_m \rrbracket \rangle}"'] & \llbracket A_1 \rrbracket \times \cdots \times \llbracket A_m \rrbracket
\end{tikzcd}$$

- An equation $\Gamma \mid t =_A u$ `pred` is interpreted as the subobject represented by the equalizer of $\llbracket \Gamma \mid t : A \rrbracket$ and $\llbracket \Gamma \mid u : A \rrbracket$:

$$\begin{tikzcd}
\llbracket \Gamma \mid t =_A u \rrbracket \arrow[r, tail] & \llbracket \Gamma \rrbracket \arrow[r, shift left, "\llbracket t \rrbracket"] \arrow[r, shift right, "\llbracket u \rrbracket"'] & \llbracket A \rrbracket
\end{tikzcd}$$

- By Proposition 3.1.9, each $\mathsf{Sub}(A)$ is a poset with binary meets. Thus we interpret a conjunction $\Gamma \mid \varphi \wedge \psi$ `pred` as the meet of subobjects

$$\llbracket \Gamma \mid \varphi \wedge \psi \rrbracket = \llbracket \Gamma \mid \varphi \rrbracket \wedge \llbracket \Gamma \mid \psi \rrbracket \;.$$

- A formula formed by weakening is interpreted as pullback along a projection:

$$\begin{tikzcd}
\llbracket \Gamma, x : A \mid \varphi \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \llbracket \Gamma \mid \varphi \rrbracket \arrow[d, tail, "i"] \\
\llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \arrow[r, "\pi"'] & \llbracket \Gamma \rrbracket
\end{tikzcd}$$

Computing this pullback one sees that the interpretation of $\llbracket \Gamma, x : A \mid \varphi \rrbracket$ turns out to be the subobject

$$\begin{tikzcd}
\llbracket \Gamma \mid \varphi \rrbracket \times \llbracket A \rrbracket \arrow[r, tail, "i \times 1_A"] & \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket
\end{tikzcd}$$

This concludes the definition of an interpretation of a cartesian theory $\mathbb{T}$ in a cartesian category $\mathcal{C}$.

As was explained in the previous section, the operation of substitution of terms into formulas is interpreted as pullback:

Lemma 3.1.12. *Let the formula $\Gamma, x : A \mid \varphi$ and the term $\Gamma \mid t : A$ be given. Then the substituted formula $\Gamma \mid \varphi[t/x]$ is interpreted as the pullback indicated in the following diagram:*

$$\begin{tikzcd}
\llbracket \Gamma \mid \varphi[t/x] \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \llbracket \Gamma, x : A \mid \varphi \rrbracket \arrow[d, tail] \\
\llbracket \Gamma \rrbracket \arrow[r, "\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle"'] & \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket
\end{tikzcd}$$

Proof. A simple induction on the structure of φ. We do the case where φ is an atomic formula $R(t_1, \dots, t_m)$. Let $\Gamma = x_1 : A_1, \dots, x_n : A_n$ and $\Gamma, x : A \mid t_i : B_i$ for $i = 1, \dots, m$, where $(B_1, \dots, B_m)$ is the signature of R. For the interpretation of $\Gamma, x : A \mid R(t_1, \dots, t_m)$, by Definition 3.1.11 we have a pullback diagram:

$$\begin{tikzcd}
\llbracket \Gamma \mid R(t_1, \dots, t_m) \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket R \rrbracket \arrow[d, tail] \\
\llbracket \Gamma, x : A \rrbracket \arrow[r, "\langle \llbracket t_1 \rrbracket, \dots, \llbracket t_m \rrbracket \rangle"'] & \llbracket B_1 \rrbracket \times \cdots \times \llbracket B_m \rrbracket
\end{tikzcd}$$

Now suppose $\Gamma \mid t : A$, and consider the substitution

$$\Gamma \mid R(t_1, \dots, t_m)[t/x] = \Gamma \mid R(t_1[t/x], \dots, t_m[t/x])$$

For the interpretations of the substituted terms $t_i[t/x]$ we have the composites

$$\llbracket t_i[t/x] \rrbracket = \llbracket t_i \rrbracket \circ \langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle : \llbracket \Gamma \rrbracket \longrightarrow \llbracket \Gamma, x : A \rrbracket \longrightarrow \llbracket B_i \rrbracket$$

by (associativity of composition and) the definition of the interpretation of terms. Thus for the interpretation of $\Gamma \mid R(t_1, \dots, t_m)[t/x]$ we have the outer pullback rectangle below.

$$\begin{tikzcd}
\llbracket \Gamma \mid R(t_1, \dots, t_m)[t/x] \rrbracket \arrow[r, dotted] \arrow[d, tail] \arrow[rr, bend left] & \llbracket \Gamma, x : A \mid R(t_1, \dots, t_m) \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket R \rrbracket \arrow[d, tail] \\
\llbracket \Gamma \rrbracket \arrow[r, "\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle"'] \arrow[rr, bend right, "\langle \llbracket t_1[t/x] \rrbracket, \dots, \llbracket t_m[t/x] \rrbracket \rangle"'] & \llbracket \Gamma, x : A \rrbracket \arrow[r, "\langle \llbracket t_1 \rrbracket, \dots, \llbracket t_m \rrbracket \rangle"'] & \llbracket B_1 \rrbracket \times \cdots \times \llbracket B_m \rrbracket
\end{tikzcd}$$

But since the righthand square is a pullback, there is a unique dotted arrow as indicated. By the 2-pullbacks lemma, the lefthand square is then also a pullback, as required. □

Exercise 3.1.13. Complete the proof.

When we deal with several different interpretations at once we may name them M, N, $\dots$, and superscript the semantic brackets accordingly, $\llbracket \Gamma \rrbracket^M$, $\llbracket \Gamma \rrbracket^N$, $\dots$

Definition 3.1.14. If $\Gamma \mid \psi \vdash \psi$ is one of the logical entailment axioms of $\mathbb{T}$ and

$$\llbracket \Gamma \mid \psi \rrbracket^M \leq \llbracket \Gamma \mid \varphi \rrbracket^M$$

holds in an interpretation M, then we say that M *satisfies* or *models* $\Gamma \mid \psi \vdash \varphi$, which we may write as

$$M \models \ (\Gamma \mid \psi \vdash \varphi) \, .$$

An interpretation M is a *model* of $\mathbb{T}$ if it satisfies all the axioms of $\mathbb{T}$.

Theorem 3.1.15 (Soundness of cartesian logic)**.** *If a cartesian theory $\mathbb{T}$ proves an entailment*

$$\Gamma \mid \psi \vdash \varphi$$

then every model M of $\mathbb{T}$ satisfies the entailment:

$$M \models \ (\Gamma \mid \psi \vdash \varphi) \ .$$

Proof. The proof proceeds by induction on the proof of the entailment. In the following we often omit the typing context Γ to simplify the notation, and all inequalities are interpreted in $\mathsf{Sub}(\llbracket \Gamma \rrbracket)$. We consider all possible last steps in the proof of the entailment:

1. Weakening: if $\llbracket \Gamma \mid \psi \rrbracket \leq \llbracket \Gamma \mid \varphi \rrbracket$ in $\mathsf{Sub}(\llbracket \Gamma \rrbracket)$ then
$$\llbracket \Gamma, x : A \mid \psi \rrbracket = \llbracket \Gamma \mid \psi \rrbracket \times A \leq \llbracket \Gamma \mid \varphi \rrbracket \times A = \llbracket \Gamma, x : A \mid \varphi \rrbracket \quad \text{in } \mathsf{Sub}(\llbracket \Gamma, x : A \rrbracket).$$

2. Substitution: by lemma 3.1.12, substitution is interpreted by pullback so that $\llbracket \varphi[t/x] \rrbracket = \langle \mathsf{1}_{\llbracket \psi \rrbracket}, \llbracket t \rrbracket \rangle^* \llbracket \varphi \rrbracket$ and $\llbracket \psi[t/x] \rrbracket = \langle \mathsf{1}_{\llbracket \psi \rrbracket}, \llbracket t \rrbracket \rangle^* \llbracket \psi \rrbracket$. Because
$$\langle \mathsf{1}_{\llbracket \psi \rrbracket}, \llbracket t \rrbracket \rangle^* : \mathsf{Sub}(\llbracket \psi \rrbracket) \to \mathsf{Sub}(\llbracket \psi \rrbracket \times \llbracket A \rrbracket)$$
is a functor it is a monotone map, therefore $\llbracket \psi \rrbracket \leq \llbracket \varphi \rrbracket$ implies
$$\langle \mathsf{1}_{\llbracket \psi \rrbracket}, \llbracket t \rrbracket \rangle^* \llbracket \psi \rrbracket \leq \langle \mathsf{1}_{\llbracket \psi \rrbracket}, \llbracket t \rrbracket \rangle^* \llbracket \varphi \rrbracket \ .$$

3. Identity: trivially
$$\llbracket \varphi \rrbracket \leq \llbracket \varphi \rrbracket \ .$$

4. Cut: if $\llbracket \psi \rrbracket \leq \llbracket \theta \rrbracket$ and $\llbracket \theta \rrbracket \leq \llbracket \varphi \rrbracket$ then clearly $\llbracket \psi \rrbracket \leq \llbracket \varphi \rrbracket$, since $\mathsf{Sub}(\llbracket \Gamma, x : A \rrbracket)$ is a poset.

5. Truth: trivially $\llbracket \psi \rrbracket \leq \llbracket \top \rrbracket$.

6. The rules for conjunction clearly hold because by the definition of infimum $\llbracket \vartheta \rrbracket \leq \llbracket \varphi \wedge \psi \rrbracket$ if, and only if, $\llbracket \vartheta \rrbracket \leq \llbracket \varphi \rrbracket$ and $\llbracket \vartheta \rrbracket \leq \llbracket \psi \rrbracket$.

7. Equality: the axiom $t =_A t$ is satisfied because an equalizer of $\llbracket t \rrbracket$ with itself is the maximal subobject:
$$\llbracket \psi \rrbracket \leq [\mathsf{1}_{\llbracket \Gamma \rrbracket} : \llbracket \Gamma \rrbracket \to \llbracket \Gamma \rrbracket] = \llbracket t =_A t \rrbracket \ .$$
For the other axiom, suppose $\llbracket \psi \rrbracket \leq \llbracket t =_A u \rrbracket$ and $\llbracket \psi \rrbracket \leq \llbracket \varphi[t/z] \rrbracket$. It suffices to show $\llbracket t =_A u \rrbracket \wedge \llbracket \varphi[t/z] \rrbracket \leq \llbracket \varphi[u/z] \rrbracket$ for then
$$\llbracket \psi \rrbracket \leq \llbracket t =_A u \rrbracket \wedge \llbracket \varphi[t/z] \rrbracket \leq \llbracket \varphi[u/z] \rrbracket \ .$$

The interpretation of $P = \llbracket t =_A u \rrbracket \wedge \llbracket \varphi[t/z] \rrbracket$ is obtained by two successive pullbacks, as in the following diagram:

$$\begin{tikzcd}
P \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket \varphi[t/z] \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket \varphi \rrbracket \arrow[d, tail] \\
\llbracket t =_A u \rrbracket \arrow[r, tail, "e"'] & \llbracket \Gamma \rrbracket \arrow[r, "\langle 1_\Gamma, \llbracket t \rrbracket \rangle"'] & \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket
\end{tikzcd}$$

Here e is the equalizer of $\llbracket u \rrbracket$ and $\llbracket t \rrbracket$. Observe that e equalizes $\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle$ and $\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket u \rrbracket \rangle$ as well:

$$\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle \circ e = \langle e, \llbracket t \rrbracket \circ e \rangle = \langle e, \llbracket u \rrbracket \circ e \rangle = \langle 1_{\llbracket \Gamma \rrbracket}, \llbracket u \rrbracket \rangle \circ e \ .$$

Therefore, if we replace $\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket t \rrbracket \rangle$ with $\langle 1_{\llbracket \Gamma \rrbracket}, \llbracket u \rrbracket \rangle$ in the above diagram, the outer rectangle still commutes. By the universal property of the pullback

$$\begin{tikzcd}
\llbracket \varphi[u/z] \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket \varphi \rrbracket \arrow[d, tail] \\
\llbracket \Gamma \rrbracket \arrow[r, "\langle 1_\Gamma, \llbracket u \rrbracket \rangle"'] & \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket
\end{tikzcd}$$

it follows that P also factors through $\llbracket \varphi[u/z] \rrbracket$, as required.

□

Example 3.1.16. Recall the cartesian theory of posets (example 3.1.7). There is one basic sort `P` and one binary relation symbol $\leq$ with signature $(\mathtt{P}, \mathtt{P})$ and the axioms of reflexivity, transitivity, and antisymmetry. A poset in a cartesian category $\mathcal{C}$ is thus given by an object P, which is the interpretation of the sort `P`, and a subobject $r : R \rightarrowtail P \times P$, which the interpretation of $\leq$, such that the axioms are satisfied. As an example we spell out when the reflexivity axiom is satisfied. The interpretation of $x : P \mid x \leq x$ is obtained by the following pullback:

$$\begin{tikzcd}
\llbracket x \leq x \rrbracket \arrow[r] \arrow[d, tail] \arrow[dr, phantom, very near start, "\lrcorner"] & R \arrow[d, tail, "r"] \\
P \arrow[r, "\Delta"'] \arrow[ur, dotted, "\rho"'] & P \times P
\end{tikzcd}$$

where $\Delta = \langle 1_P, 1_P \rangle$ is the diagonal. The first axiom is satisfied when $\llbracket x \leq x \rrbracket = 1_P$, which happens if, and only if, Δ factors through r, as indicated. Therefore, reflexivity can be expressed as follows: there exists a "reflexivity" morphism $\rho : P \to R$ such that $r \circ \rho = \Delta$. Equivalently, the morphisms $\pi_0 \circ r$ and $\pi_1 \circ r$ have a common right inverse ρ.

As an example, of a poset in a cartesian category other than Set, observe that since the definition is stated entirely in terms of finite limits, and these are computed pointwise in functor categories $\mathsf{Set}^{\mathbb{C}}$, it follows that a poset P in $\mathsf{Set}^{\mathbb{C}}$ is the same thing as a functor $P : \mathbb{C} \to \mathsf{Poset}$. Indeed, as was the case for algebraic theories, we have an equivalence (an isomorphism, actually) of categories,

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

Exercise 3.1.17. An *ordered group* is a group $(G, \cdot, i, e)$ equipped with a partial ordering $x \leq y$ that is compatible with the group multiplication, in the sense that $x \leq y$ implies $x \cdot z \leq y \cdot z$ and $z \cdot x \leq z \cdot y$. Is this the same thing as a group in the category of posets? A poset in the category of groups?

Subtypes

Let us consider whether the theory of a category is a cartesian theory. We begin by expressing the definition of a category so that it can be interpreted in any cartesian category $\mathcal{C}$. An *internal category* in $\mathcal{C}$ consists of an *object of morphisms* C_1, an *object of objects* C_0, and *domain*, *codomain*, and *identity* morphisms,

$$\mathsf{dom} : C_1 \to C_0 \;, \qquad \mathsf{cod} : C_1 \to C_0 \;, \qquad \mathsf{id} : C_0 \to C_1 \;.$$

There is also a *composition* morphism $c : C_2 \to C_1$, where C_2 is obtained by the pullback

$$\begin{tikzcd}
C_2 \arrow[r, "p_1"] \arrow[d, "p_0"'] \arrow[dr, phantom, very near start, "\lrcorner"] & C_1 \arrow[d, "\mathsf{dom}"] \\
C_1 \arrow[r, "\mathsf{cod}"'] & C_0
\end{tikzcd}$$

The following equations must hold:

$$\begin{gathered}
\mathsf{dom} \circ i = \mathsf{1}_{C_0} = \mathsf{cod} \circ i \;, \\
\mathsf{cod} \circ p_1 = \mathsf{cod} \circ c \;, \qquad \mathsf{dom} \circ p_0 = \mathsf{dom} \circ c \;. \\
c \circ \langle \mathsf{1}_{C_1}, i \circ \mathsf{dom} \rangle = \mathsf{1}_{C_1} = c \circ \langle i \circ \mathsf{cod}, \mathsf{1}_{C_1} \rangle \;,
\end{gathered}$$

The first two equations state that the domain and codomain of an identity morphism $\mathsf{1}_A$ are both A. The second equation states that $\mathsf{cod}\,(f \circ g) = \mathsf{cod}\, f$ and the third one that $\mathsf{dom}\,(f \circ g) = \mathsf{dom}\, g$. The fourth equation states that $f \circ \mathsf{1}_{\mathsf{dom}\, f} = f = \mathsf{1}_{\mathsf{cod}\, f} \circ f$. It remains to express associativity of composition. For this purpose we construct the pullback

$$\begin{tikzcd}
C_3 \arrow[r, "q_2"] \arrow[d, "q_{01}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & C_1 \arrow[d, "\mathsf{dom}"] \\
C_2 \arrow[r, "\mathsf{cod} \circ p_1"'] & C_0
\end{tikzcd}$$

The object C_3 can be thought of as the set of triples of morphisms (f, g, h) such that $\mathsf{cod}\, f = \mathsf{dom}\, g$ and $\mathsf{cod}\, g = \mathsf{dom}\, h$. We denote $q_0 = p_0 \circ q_{01}$ and $q_1 = p_1 \circ q_{01}$. The morphisms $q_0, q_1, q_2 : C_3 \to C_1$ are like three projections which select the first, second, and third element of a triple, respectively. With this notation we can write $q_{01} = \langle q_0, q_1 \rangle_{C_2}$ because q_{01} is the unique morphism such that $p_0 \circ q_{01} = q_0$ and $p_1 \circ q_{01} = q_1$. The subscript C_2 reminds us that the "pair" $\langle q_0, q_1 \rangle_{C_2}$ is obtained by the universal property of the pullback C_2.

Morphisms $c \circ q_{01} : C_3 \to C_1$ and $q_2 : C_3 \to C_1$ factor through the pullback C_2 because

$$\mathsf{cod} \circ c \circ q_{01} = \mathsf{cod} \circ p_1 \circ q_0 = \mathsf{dom} \circ q_2 \;.$$

Thus let $r : C_3 \to C_2$ be the unique factorization for which $p_0 \circ r = c \circ q_{01}$ and $p_1 \circ r = q_2$. Because p_0 and p_1 are like projections from C_2 to C_1, morphism r can be thought of as a pair of morphisms, so we write $r = \langle c \circ q_{01}, q_2 \rangle_{C_2}$. Morphism $c \circ \langle c \circ q_{01}, q_2 \rangle_{C_2} : C_3 \to C_1$ corresponds to the operations $\langle f, g, h \rangle \mapsto (f, g) \circ h$, whereas the morphism corresponding to $\langle f, g, h \rangle \mapsto f \circ (g \circ h)$ is obtained in a similar way and is equal to

$$c \circ \langle q_0, c \circ \langle q_1, q_2 \rangle_{C_2} \rangle_{C_2} : C_3 \to C_1 \;.$$

Thus associativity is expressed by the equation

$$c \circ \langle c \circ \langle q_0, q_1 \rangle_{C_2}, q_2 \rangle_{C_2} = c \circ \langle q_0, c \circ \langle q_1, q_2 \rangle_{C_2} \rangle_{C_2} \;.$$

Example 3.1.18. An internal category in Set is a small category.

Example 3.1.19. An internal category in $\mathsf{Set}^{\mathbb{C}}$ is a functor $\mathbb{C} \to \mathsf{Cat}$. Indeed, as in previous examples of cartesian theories we have an equivalence of categories,

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

We have successfully formulated the theory of a category so that it makes sense in any cartesian category. In fact, the definition of an internal category refers only to certain pullbacks, hence the notion of an internal category makes sense in any category with pullbacks. However, if we try to formulate it as a multi-sorted cartesian theory, there is a problem. Obviously, there ought to be a basic sort of objects $\mathtt{C}_0$ and a basic sort of morphisms $\mathtt{C}_1$. There are also basic function symbols with signatures

$$\mathtt{dom} : (\mathtt{C}_1; \mathtt{C}_0) \qquad\qquad \mathtt{cod} : (\mathtt{C}_1; \mathtt{C}_0) \qquad\qquad \mathtt{id} : (\mathtt{C}_0, \mathtt{C}_1) \;.$$

However, it is not clear what the signature for composition should be. It is not $(\mathtt{C}_1, \mathtt{C}_1; \mathtt{C}_1)$ because composition is undefined for non-composable pairs of morphisms. We might be tempted to postulate another basic sort $\mathtt{C}_2$ but then we would have no way of stating that $\mathtt{C}_2$ is the pullback of $\mathtt{dom}$ and $\mathtt{cod}$. And even if we somehow axiomatized the fact that $\mathtt{C}_2$ is a pullback, we would then still have to formalize the object $\mathtt{C}_3$ of composable triples, $\mathtt{C}_4$

of composable quadruples, and so on. What we lack is the ability to define the type $\mathtt{C}_2$ as a *subtype* of $\mathtt{C}_1 \times \mathtt{C}_1$.

One way to remedy the situation is to use a richer underlying type theory; in Chapter **??** we will consider the system of *dependent type theory*, which provides the means to capture such notions as the theory of categories (and much more). Here we consider a small step in that direction, namely *simple subtypes*. The formation rule for simple subtypes is

$$\frac{x : A \mid \varphi \ \mathtt{pred}}{\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}} \ \mathtt{type}}$$

We can think of $\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$ as the subobject of all those $x : A$ that satisfy φ. Note that we did not allow an arbitrary context Γ to be present. This means that we cannot define subtypes that depend on parameters, which why they are called “simple”.

Inference rules for subtypes are as follows:

$$\frac{\Gamma \mid t : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}}{\Gamma \mid \mathtt{in}_\varphi\, t : A} \qquad \frac{\Gamma \mid t : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}}{\Gamma \mid \cdot \vdash \varphi[\mathtt{in}_\varphi\, t/x]} \qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \cdot \vdash \varphi[t/x]}{\Gamma \mid \mathtt{rs}_\varphi\, t : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}}$$

$$\begin{array}{c} \Gamma, x : A \mid \varphi, \psi \vdash \theta \\ \hline\hline \Gamma, y : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}} \mid \psi[\mathtt{in}_\varphi\, y/x] \vdash \theta[\mathtt{in}_\varphi\, y/x] \end{array}$$

The first rule states that a term t of subtype $\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$ can be converted to a term $\mathtt{in}_\varphi\, t$ of type A. We can think of the constant $\mathtt{in}_\varphi$ as the *inclusion* $\mathtt{in}_\varphi : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}} \to A$. The second rule states that every term of a subtype $\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$ satisfies the defining predicate φ. The third rule states that a term t of type A which satisfies φ can be converted to a term $\mathtt{rs}_\varphi\, t$ of type $\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$. A good way to think of the constant $\mathtt{rs}_\varphi$ is as a partially defined *restriction*, or a type-casting operations, $\mathtt{rs}_\varphi : A \rightharpoonup \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$.[5] The last rule tells us how to replace a variable x of type A and an assumption φ about it with a variable y of type $\mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}$ and remove the assumption. Note that this is a two-way rule.

There are two more axioms that relate inclusions and restrictions:

$$\frac{\Gamma \mid t : \mathtt{\{}x : A \mathrel{\mathtt{|}} \varphi\mathtt{\}}}{\Gamma \mid \cdot \vdash \mathtt{rs}_\varphi\, (\mathtt{in}_\varphi\, t) = t} \qquad\qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \cdot \vdash \varphi[t/x]}{\Gamma \mid \cdot \vdash \mathtt{in}_\varphi\, (\mathtt{rs}_\varphi\, t) = t\ .}$$

In an informal discussion it is customary for the inclusions and restrictions to be omitted, or at least for the subscript φ to be missing.[6]

Exercise 3.1.20. Suppose $x : A \mid \psi$ and $x : A \mid \varphi$ are formulas. Show that

$$x : A \mid \psi \vdash \varphi$$

[5] Inclusions and restrictions are like type-casting operations in some programming languages. For example in Java, an inclusion corresponds to an (implicit) type cast from a class to its superclass, whereas a restriction corresponds to a type cast from a class to a subclass. Must I write that Java is a registered trademark of Sun Microsystems?

[6] Strictly speaking, even the notation $\mathtt{in}_\varphi\, t$ is imprecise because it does not indiciate that ϕ stands in the context $x : A$. The correct notation would be $\mathtt{in}_{(x:A|\varphi)}\, t$, where x is bound in the subscript. A similar remark holds for $\mathtt{rs}_\varphi\, t$.

is provable if, and only if, $\{x : A \mid \psi\}$ factors through $\{x : A \mid \varphi\}$, which means that there exists a term k,

$$y : \{x : A \mid \psi\} \mid k : \{x : A \mid \varphi\} \ ,$$

such that

$$y : \{x : A \mid \psi\} \mid \cdot \vdash \mathtt{in}_\psi\, y =_A \mathtt{in}_\varphi\, k$$

is provable. Show also that k is determined uniquely up to provable equality.

Example 3.1.21. We are now able to formulate the theory of a category as a cartesian theory whose underlying type theory has product types and subset types. The basic types are the type of objects $\mathtt{C}_0$ and the type of morphisms $\mathtt{C}_1$. We define the type $\mathtt{C}_2$ to be

$$\mathtt{C}_2 \equiv \{p : C_1 \times C_1 \mid \mathtt{cod}(\mathtt{fst}\, p) = \mathtt{dom}(\mathtt{snd}\, p)\} \ .$$

The basic function symbols and their signatures are:

$$\mathtt{dom} : \mathtt{C}_1 \to \mathtt{C}_0 \ , \qquad \mathtt{cod} : \mathtt{C}_1 \to \mathtt{C}_0 \ , \qquad \mathtt{id} : \mathtt{C}_0 \to \mathtt{C}_1 \ , \qquad \mathtt{c} : \mathtt{C}_2 \to \mathtt{C}_1 \ .$$

The axioms are:

$$\begin{aligned}
a : \mathtt{C}_0 \mid \cdot &\vdash \mathtt{dom}(\mathtt{id}(a)) = a \\
a : \mathtt{C}_0 \mid \cdot &\vdash \mathtt{cod}(\mathtt{id}(a)) = a \\
f : \mathtt{C}_1, g : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g) &\vdash \mathtt{dom}(\mathtt{c}(\mathtt{rs}\,\langle f, g\rangle)) = f \\
f : \mathtt{C}_1, g : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g) &\vdash \mathtt{cod}(\mathtt{c}(\mathtt{rs}\,\langle f, g\rangle)) = g \\
f : \mathtt{C}_1 \mid \cdot &\vdash \mathtt{c}(\mathtt{rs}\,\langle \mathtt{id}(\mathtt{dom}(f)), f\rangle) = f \\
f : \mathtt{C}_1 \mid \cdot &\vdash \mathtt{c}(\mathtt{rs}\,\langle f, \mathtt{id}(\mathtt{cod}(f))\rangle) = f
\end{aligned}$$

Lastly, the associativity axiom is

$$\begin{aligned}
&f : \mathtt{C}_1, g : \mathtt{C}_1, h : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g), \mathtt{cod}(g) = \mathtt{dom}(h) \vdash \\
&\qquad\qquad \mathtt{c}(\mathtt{rs}\,\langle \mathtt{c}(\mathtt{rs}\,\langle f, g\rangle), h\rangle) = \mathtt{c}(\mathtt{rs}\,\langle f, \mathtt{c}(\mathtt{rs}\,\langle g, h\rangle)\rangle) \ .
\end{aligned}$$

This notation is quite unreadable. If we write $g \circ f$ instead of $\mathtt{c}(\mathtt{rs}\,\langle f, g\rangle)$ then the axioms take on a more familiar form. For example, associativity is just $h \circ (g \circ f) = (h \circ g) \circ f$. However, we need to remember that we may form the term $g \circ f$ only if we first prove $\mathtt{dom}(g) = \mathtt{cod}(f)$.

A subtype $\{x : A \mid \varphi\}$ is interpreted as the domain of a monomorphism representing $x : A \mid \varphi$:

$$\begin{tikzcd}
\llbracket \{x : A \mid \varphi\} \rrbracket \arrow[r, tail, "\llbracket x : A \mid \varphi \rrbracket"] & \llbracket A \rrbracket
\end{tikzcd}$$

Some care must be taken here because monos representing a given subobject are only determined up to isomorphism. We assume that a suitable canonical choice of monos can be made.

An inclusion $\Gamma \mid \mathtt{in}_{\varphi}\, t : A$ is interpreted as the composition

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, "\llbracket t \rrbracket"] & \llbracket \mathtt{\{}x : A \mid \varphi\mathtt{\}} \rrbracket \arrow[r, tail, "\llbracket x : A \mid \varphi \rrbracket"] & \llbracket A \rrbracket
\end{tikzcd}$$

A restriction $\Gamma \mid \mathtt{rs}_{\varphi}\, t : \mathtt{\{}x : A \mid \varphi\mathtt{\}}$ is interpreted as the unique $\overline{\llbracket t \rrbracket}$ which makes the following diagram commute:

$$\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[r, "\overline{\llbracket t \rrbracket}"] \arrow[dr, "\llbracket t \rrbracket"'] & \llbracket x : A \mid \varphi \rrbracket \arrow[d, tail] \\
& \llbracket A \rrbracket
\end{tikzcd}$$

Exercise 3.1.22. Formulate and prove a soundness theorem for subtypes. Pay attention to the interpretation of restrictions, where you need to show unique existence of $\overline{\llbracket t \rrbracket}$.

Remark 3.1.23. Another approach to the logic of cartesian categories that captures the theory of categories and related notions involving partial operations is that of *essentially algebraic theories*, due to P. Freyd; see [Fre72, PV07]. A third approach is that of *dependent type theory* to be developed in 5.1 below. Finally, we will see in Section 3.2.3 that the theory of categories can be formulated as a *regular theory*.

3.1.4 Quantifiers as adjoints

The categorical semantics of quantification is one of the central features of the subject, and quite possibly one of the nicest contributions of categorical logic to the field of logic. You might expect that the quantifiers $\forall$ and $\exists$ are "just a big conjunction and disjunction", respectively. In fact the Polish school of algebraic logic worked to realize this point of view—but categorical logic shows how quantifiers can be treated algebraically as adjoint functors, giving a more satisfactory theory that generalizes to categories in which the subobject lattices are not (co)complete. The original treatment can be found in the classic paper [**?**].

Let us first recall the rules of inference for quantifiers. The formation rules are:

$$\frac{\Gamma, x : A \mid \varphi\ \mathtt{pred}}{\Gamma \mid (\exists\, x : A \,.\, \varphi)\ \mathtt{pred}} \qquad\qquad \frac{\Gamma, x : A \mid \varphi\ \mathtt{pred}}{\Gamma \mid (\forall\, x : A \,.\, \varphi)\ \mathtt{pred}}$$

The variable x is bound in $\forall\, x : A \,.\, \varphi$ and $\exists\, x : A \,.\, \varphi$. If x and y are distinct variables and x does not occur freely in the term t then substitution of t for y commutes with quantification over x:

$$\begin{aligned}
(\exists\, x : A \,.\, \varphi)[t/y] &= \exists\, x : A \,.\, (\varphi[t/y]) \ , \\
(\forall\, x : A \,.\, \varphi)[t/y] &= \forall\, x : A \,.\, (\varphi[t/y]) \ .
\end{aligned} \tag{3.2}$$

For each quantifier we have a two-way rule of inference:

$$\begin{array}{c} \Gamma, x : A \mid \varphi \vdash \vartheta \\ \hline\hline \Gamma \mid (\exists\, x : A \,.\, \varphi) \vdash \vartheta \end{array} \qquad\qquad \begin{array}{c} \Gamma, x : A \mid \psi \vdash \varphi \\ \hline\hline \Gamma \mid \psi \vdash \forall\, x : A \,.\, \varphi \end{array}$$

Note that these rules implicitly impose the usual condition that x must not occur freely in ψ and ϑ, because ψ and ϑ are supposed to be well formed in context Γ, which does not contain x.

Exercise 3.1.24. A common way of stating the inference rules for quantifiers is as follows. For the universal quantifier, the introduction and elimination rules are

$$\frac{\Gamma, x : A \mid \psi \vdash \varphi}{\Gamma \mid \psi \vdash \forall\, x : A \,.\, \varphi} \qquad\qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \psi \vdash \forall\, x : A \,.\, \varphi}{\Gamma \mid \psi \vdash \varphi[t/x]}$$

The introduction rule for existential quantifier is

$$\frac{\Gamma \mid t : A \qquad \Gamma \mid \psi \vdash \varphi[t/x]}{\Gamma \mid \psi \vdash \exists\, x : A \,.\, \varphi}$$

and the elimination rule is

$$\frac{\Gamma \mid \psi \vdash \exists\, x : A \,.\, \varphi \qquad \Gamma, x : A \mid \varphi \vdash \vartheta}{\Gamma \mid \psi \vdash \vartheta}$$

Note that these rules implicitly impose a requirement that x does not occur in Γ and that it does not occur freely in ψ because the context $\Gamma, x : A$ must be well formed and the hypotheses ψ must be well formed in context Γ. Show that these rules can be derived from the ones above, and vice versa. Of course, you may also use the inference rules for cartesian logic, cf. page 108.

In order to discover what the semantics of existential quantifier ought to be, we look at the following instance of the two-way rule for quantifiers:

$$\begin{array}{c} y : B, x : A \mid \varphi \vdash \vartheta \\ \hline\hline y : B \mid \exists\, x : A \,.\, \varphi \vdash \vartheta \end{array} \tag{3.3}$$

First observe that this rule implicitly requires

$$y : B, x : A \mid \varphi \ \mathtt{pred} \qquad\qquad y : B \mid \vartheta \ \mathtt{pred} \qquad\qquad y : B \mid (\exists\, x : A \,.\, \varphi) \ \mathtt{pred}$$

This is required for the entailments to be well-formed. The fourth judgement

$$y : B, x : A \mid \vartheta \ \mathtt{pred}$$

follows from the second one above by weakening,

$$\frac{y : B \mid \vartheta \ \mathtt{pred}}{y : B, x : A \mid \vartheta \ \mathtt{pred}}$$

The interpretations of φ, ϑ, and $\exists\, x : A \,.\, \varphi$ are therefore subobjects

$$\begin{gathered}
[\![y : B, x : A \mid \varphi]\!] \in \mathsf{Sub}([\![B]\!] \times [\![A]\!])\ , \\
[\![y : B \mid \vartheta]\!] \in \mathsf{Sub}([\![B]\!])\ , \\
[\![y : B \mid \exists\, x : A \,.\, \varphi]\!] \in \mathsf{Sub}([\![B]\!])\ .
\end{gathered}$$

And the weakened instance of ϑ in the context $y : B, x : A$ is interpreted by pullback along a projection, cf. page 112, as in the following pullback diagram:

$$\begin{tikzcd}
{[\![y : B, x : A \mid \vartheta]\!]} \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & {[\![B]\!] \times [\![A]\!]} \arrow[d, "\pi"] \\
{[\![y : B \mid \vartheta]\!]} \arrow[r, tail] & {[\![B]\!]}
\end{tikzcd}$$

Thus we have

$$[\![y : B, x : A \mid \vartheta]\!] = \pi^* [\![y : B \mid \vartheta]\!]\, ,$$

with weakening interpreted as the pullback functor

$$\pi^* : \mathsf{Sub}([\![B]\!]) \to \mathsf{Sub}([\![B]\!] \times [\![A]\!])\, .$$

We will interpret existential quantification $\exists x : A$ as a suitable functor

$$\exists_A : \mathsf{Sub}([\![B]\!] \times [\![A]\!]) \to \mathsf{Sub}([\![B]\!])$$

so that

$$[\![y : B \mid \exists\, x : A \,.\, \varphi]\!] = \exists_A [\![y : B, x : A \mid \varphi]\!]\ .$$

The interpretation of the two-way rule (3.3) then becomes a two-way inequality rule

$$\frac{[\![y : B, x : A \mid \varphi]\!] \leq \pi^* [\![y : B \mid \vartheta]\!]}{\exists_A [\![y : B, x : A \mid \varphi]\!] \leq [\![y : B \mid \vartheta]\!]}$$

Replacing the interpretations of φ and ϑ by general subobjects $S \in \mathsf{Sub}([\![B]\!] \times [\![A]\!])$ and $T \in \mathsf{Sub}([\![B]\!])$, we obtain the more suggestive formulation

$$\frac{S \leq \pi^* T}{\exists_A S \leq T} \tag{3.4}$$

This is of course nothing but an adjunction between $\exists_A$ and π^*. Indeed, the operations $\exists_A$ and π^* are functors on the posets of subjects $\mathsf{Sub}([\![B]\!] \times [\![A]\!])$ and $\mathsf{Sub}([\![B]\!])$, and the bijection of hom-sets (3.4) is exactly the statement of an adjunction between them. Thus *existential quantification is left-adjoint to weakening*:

$$\exists_A \dashv \pi^*$$

An exactly dual argument shows that *universal quantification is right-adjoint to weakening*:

$$\pi^* \dashv \forall_A$$

Thus, in sum, we have that the rules of inference require that the quantifiers be interpreted as operations adjoint to the interpretation of weakening, i.e. pullback π^* along the projection $\pi : \llbracket B \rrbracket \times \llbracket A \rrbracket \to \llbracket B \rrbracket$.

$$\begin{tikzcd}[row sep=huge] \mathsf{Sub}(\llbracket B \rrbracket \times \llbracket A \rrbracket) \arrow[d, bend right=60, "\exists"'] \arrow[d, bend left=60, "\forall"] \arrow[d, phantom, bend right=25, "\dashv"] \arrow[d, phantom, bend left=25, "\dashv"] \\ \mathsf{Sub}(\llbracket B \rrbracket) \arrow[u, "\pi^*" description] \end{tikzcd}$$

Note that the familiar side-conditions on the conventional rules for the quantifiers, to the effect that "x cannot occur freely in ψ", etc., which may seem like tiresome book-keeping, are actually of the essence, since they actually express the weakening operation to which the quantifiers themselves are adjoints.

Let us see how this works for the usual interpretation in Set. A predicate $y : B, x : A \mid \varphi$ corresponds to a subset $\Phi \subseteq B \times A$, and $y : B \mid \vartheta$ corresponds to a subset $\Theta \subseteq B$. Weakening of Θ is the subset $\pi^*\Theta = \Theta \times A \subseteq B \times A$. Then we have

$$\begin{aligned} \exists_A \Phi &= \big\{ y \in B \;\big|\; \exists\, x : A \,.\, \langle x, y \rangle \in \Phi \big\} \subseteq B \;, \\ \forall_A \Phi &= \big\{ y \in B \;\big|\; \forall\, x : A \,.\, \langle x, y \rangle \in \Phi \big\} \subseteq B \;. \end{aligned}$$

A moment's thought convinces us that with this interpretation we do indeed have

$$\begin{array}{c} \Phi \subseteq \Theta \times A \\ \hline\hline \exists_A \Phi \subseteq \Theta \end{array} \qquad\qquad \begin{array}{c} \Theta \times A \subseteq \Phi \\ \hline\hline \Theta \subseteq \forall_A \Phi \end{array}$$

The unit of the adjunction $\exists_A \dashv \pi^*$ amounts to the inequality

$$\Phi \subseteq (\exists_A \Phi) \times A \;, \tag{3.5}$$

and the universal property of the unit says that $\exists_A \Phi$ is the smallest set satisfying (3.5). Similarly, the counit of the adjunction $\pi^* \dashv \forall_A$ is just the inequality

$$(\forall_A \Phi) \times A \subseteq \Phi \;, \tag{3.6}$$

and the universal property of the counit says that $\forall_A \Phi$ is the largest set satisfying (3.6). Figure 3.1 shows the geometric meaning of existential and universal quantification.

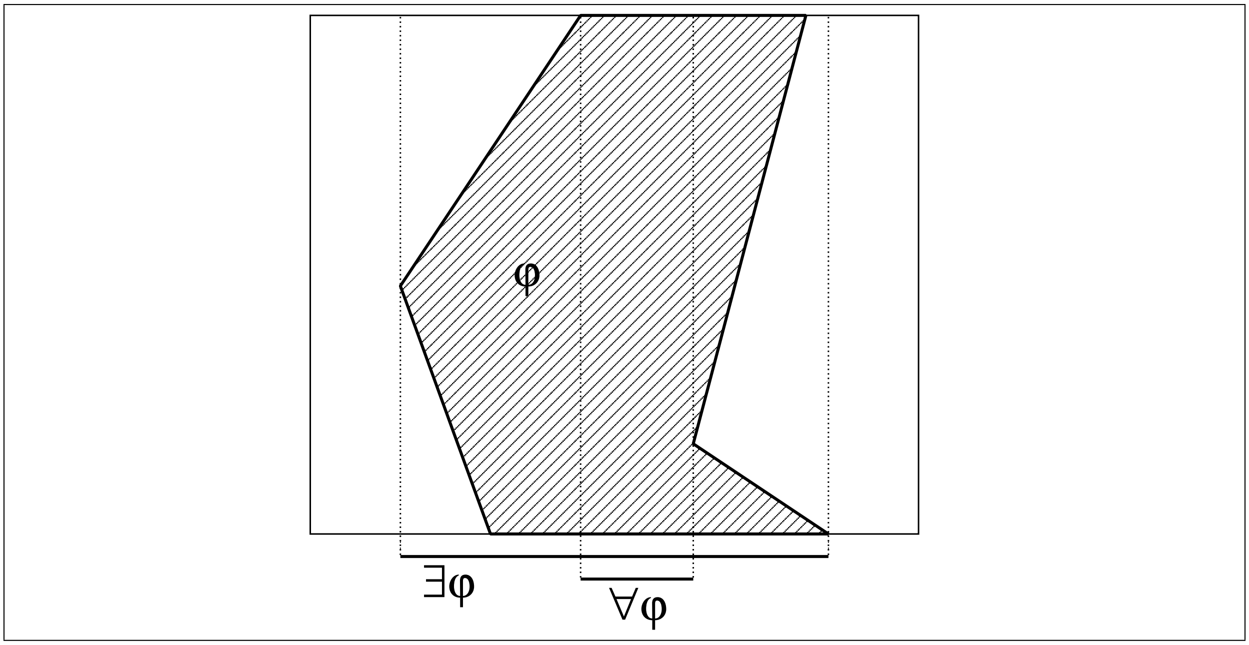


Figure 3.1: $\exists\varphi$ and $\forall\varphi$

Exercise 3.1.25. What do the universal properties of the counit of $\exists_A \dashv \pi^*$ and the unit of $\pi^* \dashv \forall_A$ say?

The weakening functor π^* is a special case of a pullback functor $f^* : \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ for a morphism $f : B \to A$. This suggests that we may regard the left and the right adjoint to f^* as a kind of generalized existential and universal quantifier. We may indeed be tempted to simply *define* the quantifiers as left and right adjoints to general pullback functors. However there is a bit more to quantifiers than that—we are still missing the important *Beck-Chevalley condition.*

The Beck-Chevalley condition

Recall from (3.2) that quantification commutes with substitution, as long as no variables are captured by the quantifier. Thus if $\Gamma \mid t : B$ and $\Gamma, y : B, x : A \mid \varphi$ `pred` then

$$\begin{aligned}(\exists\, x : A \,.\, \varphi)[t/y] &= \exists\, x : A \,.\, (\varphi[t/y]) \ , \\ (\forall\, x : A \,.\, \varphi)[t/y] &= \forall\, x : A \,.\, (\varphi[t/y]) \ .\end{aligned}$$

If the semantics of quantification is to be sound, the interpretation of these equations must be valid. Because substitution of a term in a formula is interpreted as pullback, this means exactly that quantifiers must be *stable* under pullbacks. This is known as the *Beck-Chevalley condition.*

Definition 3.1.26. A family of functors $F_f : \mathsf{Sub}(A) \to \mathsf{Sub}(B)$ parametrized by morphisms $f : A \to B$ is said to satisfy the *Beck-Chevalley condition* when for every pullback as on the left-hand side, the right-hand square commutes:

$$\begin{tikzcd}
C \arrow[r, "h"] \arrow[d, "k"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\
D \arrow[r, "g"'] & B
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathsf{Sub}(C) \arrow[d, "F_k"'] & \mathsf{Sub}(A) \arrow[l, "h^*"'] \arrow[d, "F_f"] \\
\mathsf{Sub}(D) & \mathsf{Sub}(B) \arrow[l, "g^*"]
\end{tikzcd}$$

To convince ourselves that Beck-Chevalley condition is what we want, we spell it out explicitly in the case of a substitution into an existentially quantified formula. In order to keep the notation simple we omit the semantic brackets $[\![-]\!]$. Suppose we have a term $\Gamma \mid t : B$ and a formula $\Gamma, y : B, x : A \mid \varphi$ `pred`. The diagram

$$\begin{tikzcd}
\Gamma \times A \arrow[r, "\langle \pi_0, t \circ \pi_0, \pi_1 \rangle"] \arrow[d, "\pi_0^{\Gamma,A}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Gamma \times B \times A \arrow[d, "\pi_0^{\Gamma,B,A}"] \\
\Gamma \arrow[r, "\langle 1_\Gamma, t \rangle"'] & \Gamma \times B
\end{tikzcd}$$

is a pullback. By the Beck-Chevalley condition for $\exists$, the following square commutes:

$$\begin{tikzcd}
\mathsf{Sub}(\Gamma \times A) \arrow[d, "\exists_A^{\Gamma,A}"'] & \mathsf{Sub}(\Gamma \times B \times A) \arrow[l, "\langle \pi_0, t \circ \pi_0, \pi_1 \rangle^*"'] \arrow[d, "\exists_A^{\Gamma,B,A}"] \\
\mathsf{Sub}(\Gamma) & \mathsf{Sub}(\Gamma \times B) \arrow[l, "\langle 1_\Gamma, t \rangle^*"]
\end{tikzcd}$$

Therefore, for $\Gamma, y : B, x : A \mid \varphi$ `pred`, we have

$$[\![(\exists\, x : A \,.\, \varphi)[t/y]]\!] = \langle 1_\Gamma, t \rangle^* (\exists_A^{\Gamma,B,A} [\![\varphi]\!]) = \\ \exists_A^{\Gamma,A} (\langle \pi_0, t \circ \pi_0, \pi_1 \rangle^* [\![\varphi]\!]) = [\![\exists\, x : A \,.\, (\varphi[t/y])]\!] \;.$$

This is indeed precisely the equation we wanted. The Beck-Chevalley condition says that (the interpretations of) the quantifiers commute with pullbacks, in just the way that the syntactic operations of applying quantifiers to formulas commute with substitutions of terms (which are interpreted as pullbacks).

Definition 3.1.27. A cartesian category $\mathcal{C}$ has *existential quantifiers* if, for every $f : A \to B$, the left adjoint $\exists_f \dashv f^*$ exists and it satisfies the Beck-Chevalley condition. Similarly, $\mathcal{C}$ has *universal quantifiers* if the right adjoints $f^* \dashv \forall_f$ exist and they satisfy the Beck-Chevalley condition.

It is convenient to know that, if we have both adjoints $\exists_f \dashv f^* \dashv \forall_f$, it actually suffices to have the Beck-Chevalley condition for either one in order to infer it for both:

Proposition 3.1.28. *If for every $f : A \to B$, both the left and right adjoints exist*

$$\exists_f \dashv f^* \dashv \forall_f$$

then the left adjoint satisfies the Beck-Chevalley condition iff the right adjoint does.

Proof. Suppose we have the Beck-Chevalley condition for the left adjoints $\exists$, and that we are given a pullback square as on the left below. We want to check the Beck-Chevalley square for the right adjoints $\forall$, as indicated on the right below.

$$\begin{tikzcd}
C \arrow[r, "h"] \arrow[d, "k"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\
D \arrow[r, "g"'] & B
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathsf{Sub}(C) \arrow[d, "\forall_k"'] & \mathsf{Sub}(A) \arrow[l, "h^*"'] \arrow[d, "\forall_f"] \\
\mathsf{Sub}(D) & \mathsf{Sub}(B) \arrow[l, "g^*"]
\end{tikzcd}$$

Swapping all the functors in the righthand diagram for their left adjoints we obtain the following.

$$\begin{tikzcd}
\mathsf{Sub}(C) \arrow[r, "\exists_h"] & \mathsf{Sub}(A) \\
\mathsf{Sub}(D) \arrow[r, "\exists_g"'] \arrow[u, "k^*"] & \mathsf{Sub}(B) \arrow[u, "f^*"']
\end{tikzcd}$$

But this is a Beck-Chevalley square for (the "transpose" of) the original pullback diagram, and therefore commutes by the Beck-Chevalley condition for the left adjoints $\exists$. The original diagram of right adjoints therefore also commutes, by uniqueness of adjoints.

The argument for the dual case is, well, dual. □

Remark 3.1.29. The counit of the adjunction for $\forall$ is $x : A \mid \forall x : A.\, \varphi \vdash \varphi$, while the unit of the $\exists$ adjunction is $x : A \mid \varphi \vdash \exists\, x : A\,.\, \varphi$. From the transitivity of $\vdash$ in any context, we therefore obtain:

$$x : A \mid \forall x : A.\, \varphi \vdash \exists x : A\,.\varphi\,. \tag{3.7}$$

If there is a term $a : 1 \to A$, we can infer $\forall x : A.\, \varphi \vdash \exists x : A\,.\varphi$ (in the empty context) by substituting it (vacuously) for $x : A$ in (3.7). The inference from $\forall$ to $\exists$, which is valid in classical predicate logic, presumes the domain of quantification is non-empty. By keeping track of the relevant contexts, our system of rules for quantifiers is also sound for domains of quantification that may not have any "global points" $a : 1 \to A$.

Exercise 3.1.30. In Set we can identify $\mathsf{Sub}(-)$ with powersets because $\mathsf{Sub}(X) \cong \mathcal{P}X$. Then quantifiers along a function $f : A \to B$ are functions

$$\exists_f : \mathcal{P}A \to \mathcal{P}B \;, \qquad\qquad \forall_f : \mathcal{P}A \to \mathcal{P}B \;.$$

Verify that

$$\begin{aligned} \exists_f U &= \big\{ b \in B \;\big|\; \exists\, a : A \,.\, (f a = b \land a \in U) \big\} \;, \\ \forall_f U &= \big\{ b \in B \;\big|\; \forall\, a : A \,.\, (f a = b \Rightarrow a \in U) \big\} \;. \end{aligned}$$

Thus $\exists_f U$ is just the usual direct image of U by f, sometimes written $f_!(U)$, or simply $f(U)$. But have you seen $\forall_f U$ before? It can also be written as $\forall_f U = \big\{ b \in B \;\big|\; f^* \{b\} \subseteq U \big\}$. What is the meaning of $\exists_q$ and $\forall_q$ when $q : A \to A/{\sim}$ is a canonical quotient map that maps an element $x \in A$ to its equivalence class $qx = [x]$ under an equivalence relation $\sim$ on A?

3.2 Regular and coherent logic

We next consider the question of when a cartesian category has existential quantifiers. It turns out that this is closely related to the notion of a *regular category*, a concept which first arose in the context of abelian categories and axiomatic homology theory, quite independently of categorical logic. We will see for instance that all algebraic categories, in the sense of Chapter **??**, are regular.

3.2.1 Regular categories

Throughout this section we work in a cartesian category $\mathcal{C}$. We begin with some general definitions. The *kernel pair* of a morphism $f : A \to B$ is the pair of morphisms $k_1, k_2 : K \rightrightarrows A$ obtained as in the following pullback

$$\begin{tikzcd} K \arrow[r, "k_2"] \arrow[d, "k_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

Note that a kernel pair determines an equivalence relation $\langle k_1, k_2 \rangle : K \rightarrowtail A \times A$, in the sense that the map $\langle k_1, k_2 \rangle$ is a mono that satisfies the reflexivity, symmetry and transitivity conditions. In Set the mono $\langle k_1, k_2 \rangle : K \rightarrowtail A \times A$ is the equivalence relation $\sim$ on A defined by

$$x \sim y \iff f x = f y \;.$$

Indeed, a kernel pair in a general cartesian category is a model of the cartesian theory of an equivalence relation, in the sense of example 3.1.8.

Exercise 3.2.1. Prove this.

In general, the *quotient* by the equivalence relation determined by the kernel pair k_1, k_2 is their coequalizer $q : A \to Q$, if it exists,

$$\begin{tikzcd}
K \arrow[r, shift left, "k_1"] \arrow[r, shift right, "k_2"'] & A \arrow[r, "q"] & Q
\end{tikzcd}$$

Such a coequalizer is called a *kernel quotient.*

Because $f \circ k_1 = f \circ k_2$, we see that f factors through q by a unique morphism $m : Q \to A$,

$$\begin{tikzcd}
K \arrow[r, shift left, "k_1"] \arrow[r, shift right, "k_2"'] & A \arrow[rr, "f"] \arrow[dr, "q"'] & & B \\
& & Q \arrow[ur, "m"'] &
\end{tikzcd} \tag{3.8}$$

As a coequalizer, $q : A \to Q$ is always epic; indeed, epis that are coequalizers will be called *regular epimorphisms* and will be denoted by arrows with triangular heads:

$$\begin{tikzcd}
e : A \arrow[r, -{Triangle[open]}] & B
\end{tikzcd}$$

It is of some interest to know when the second factor $m : Q \to B$ in (3.8) is guaranteed to be a mono. For example, in Set the function $m : Q \to B$ is defined by $m[x] = fx$, where $Q = A/{\sim}$ as above. In this case m is indeed injective, because $m[x] = m[y]$ implies $fx = fy$, hence $x \sim y$ and $[x] = [y]$.

Definition 3.2.2. A category with finite limits is *regular* when it has kernel quotients, and regular epis are stable under pullback. Thus, in detail:

1. the kernel pair of any map has a coequalizer, and
2. any pullback of a regular epi is a regular epi.

Exercise 3.2.3. Suppose $e : A \twoheadrightarrow B$ is a regular epi. Prove that it is the coequalizer of its own kernel pair.

Let us return to (3.8) and show that m is monic in any regular category. Consider the following diagram, in which h_1, h_2 are constructed as the kernel pair of m, and the other three squares are constructed as pullbacks:

$$\begin{tikzcd}
K \arrow[r, -{Triangle[open]}, "p_2"] \arrow[d, -{Triangle[open]}, "p_1"'] \arrow[dr, -{Triangle[open]}, "r"] \arrow[dr, phantom, very near start, "\lrcorner"] & \cdot \arrow[r] \arrow[d, -{Triangle[open]}, "s_2"] \arrow[dr, phantom, very near start, "\lrcorner"] & A \arrow[d, -{Triangle[open]}, "q"] \\
\cdot \arrow[r, -{Triangle[open]}, "s_1"'] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & H \arrow[r, "h_2"] \arrow[d, "h_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & Q \arrow[d, "m"] \\
A \arrow[r, -{Triangle[open]}, "q"'] & Q \arrow[r, "m"'] & B
\end{tikzcd}$$

Because all the smaller squares are pullbacks the large square is a pullback as well, therefore the left-hand vertical morphism is $k_1 : K \to A$, and the morphism across the top is $k_2 : K \to A$, and we have the kernel pair $k_1, k_2 : K \rightrightarrows A$ of $f = m \circ q$. The morphisms s_1, s_2, p_1, and p_2 are all regular epis because they are pullbacks of the regular epi q. The morphism $r = s_2 \circ p_2 = s_1 \circ p_1$ is epic because it is a composition of regular epis. Observe that

$$h_1 \circ r = q \circ k_1 = q \circ k_2 = h_2 \circ r \ ,$$

and so, because r is epic, $h_1 = h_2$. But this means that m is monic, since the maps in its kernel pair are equal; indeed, given any $u, v : U \to Q$ with $m \circ u = m \circ v$, there exists a $w : U \to H$ such that $u = w \circ h_1 = w \circ h_2 = v$.

Proposition 3.2.4. *In a regular category every morphism $f : A \to B$ factors as a composition of a regular epi q followed by a mono m,*

$$\begin{tikzcd}
A \arrow[r, -{Triangle[open]}, "q"'] \arrow[rr, bend left, "f"] & Q \arrow[r, tail, "m"'] & B
\end{tikzcd}$$

The factorization is unique up to isomorphism.

Proof. By uniqueness of the factorization we mean that if

$$\begin{tikzcd}
A \arrow[r, -{Triangle[open]}, "q'"'] \arrow[rr, bend left, "f"] & Q' \arrow[r, tail, "m'"'] & B
\end{tikzcd}$$

is another such factorization, then there exists an isomorphism $i : Q \to Q'$ such that $q' = i \circ q$ and $m = m' \circ i$.

$$\begin{tikzcd}
A \arrow[r, -{Triangle[open]}, "q'"] \arrow[d, -{Triangle[open]}, "q"'] & Q' \arrow[d, tail, "m'"] \\
Q \arrow[r, tail, "m"'] \arrow[ur, dotted, "i"'] & B
\end{tikzcd}$$

As the factorization of f we take the one constructed in (3.8). Then q is a regular epi by construction, and we have just shown that m is monic. So it only remains to show that the factorization is unique. Suppose f also factors as $f = m' \circ q'$ where q' is a regular epi and m' is monic. Consider the following diagram, in which k_1, k_2 is the kernel pair of f, q is the coequalizer of k_1 and k_2, and h_1, h_2 is the kernel pair of q' so that q' is the coequalizer of h_1 and h_2:

$$\begin{tikzcd}
& H \arrow[d, shift right, "h_1"'] \arrow[d, shift left, "h_2"] & \\
K \arrow[r, shift left, "k_1"] \arrow[r, shift right, "k_2"'] & A \arrow[r, two heads, "q"] \arrow[d, two heads, "q'"'] & Q \arrow[d, tail, "m"] \arrow[dl, dotted, shift right, "i"'] \\
& Q' \arrow[r, tail, "m'"'] \arrow[ur, dotted, shift right, "j"'] & B
\end{tikzcd}$$

Because $m' \circ q' \circ k_1 = m \circ q \circ k_1 = m \circ q \circ k_2 = m' \circ q' \circ k_2$ and m' is monic, $q' \circ k_1 = q' \circ k_2$. So there exists a unique $i : Q \to Q'$ such that $q' = i \circ q$. But then $m' \circ i \circ q = m' \circ q' = f = m \circ q$ and because q is epi, $m' \circ i = m$.

We prove that i is iso by constructing its inverse j. Because $m \circ q \circ h_1 = m' \circ q \circ h_1 = m' \circ q \circ h_2 = m \circ q \circ h_2$ and m is monic, $q \circ h_1 = q \circ h_2$. So there exists a unique $j : Q' \to Q$ such that $q = j \circ q'$. Now we have $i \circ j \circ q' = i \circ q = 1_{Q'} \circ q'$, from which we conclude that $i \circ j = 1_{Q'}$ because q' is epi. Similarly, $j \circ i \circ q = j \circ q' = 1_Q \circ q$, therefore $j \circ i = 1_Q$. □

Corollary 3.2.5. *A map $f : A \to B$ that is both a regular epi and a mono is an iso.*

Proof. Consider the following outer square, regarded as two different reg-epi/mono factorizations.

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

A diagonal d is then an inverse of f. □

A factorization $f = m \circ q$ as in Proposition 3.2.4 determines a subobject

$$\mathsf{im}(f) = [m : Q \rightarrowtail B] \in \mathsf{Sub}(B) \ ,$$

called the *image of f*. It is characterized as the least subobject of B through which f factors.

Proposition 3.2.6. *For a morphism $f : A \to B$ in a regular category $\mathcal{C}$, the image $\mathsf{im}(f) \rightarrowtail B$ is the least subobject $U \rightarrowtail B$ of B through which f factors.*

Proof. Suppose f factors through $v : V \rightarrowtail B$ as

$$\begin{tikzcd}
A \arrow[r, "g"'] \arrow[rr, bend left, "f"] & V \arrow[r, tail, "v"'] & B
\end{tikzcd}$$

and consider the factorization of f, as in (3.8). Since $v \circ g \circ k_1 = f \circ k_1 = f \circ k_2 = v \circ g \circ k_2$ and v is mono, $g \circ k_1 = g \circ k_2$, therefore there exists a unique $\overline{g} : Q \to V$ such that $g = \overline{g} \circ q$. Now $v \circ \overline{g} \circ q = v \circ g = f = m \circ q$ and because q is epic, $v \circ \overline{g} = m$ as required. (The reader should draw the corresponding diagram.) □

Definition 3.2.7. A functor $F : \mathcal{C} \to \mathcal{D}$ is *regular* if it preserves finite limits and regular epis. It follows that F preserves image factorizations. The category of regular functors $\mathcal{C} \to \mathcal{D}$ and natural transformations is denoted by $\mathsf{Reg}(\mathcal{C}, \mathcal{D})$.

Examples of regular categories

Let us consider some examples of regular categories.

1. The category Set is regular. It is complete and cocomplete, so it has in particular all finite limits and coequalizers. To show that the pullback of a regular epi is again a regular epi, note that in Set the epis are exactly the surjections, and a surjection is a quotient of its kernel pair, and thus a regular epi. It therefore it suffices to show that the pullback of a surjection is a surjection, which is easy.

2. More generally, any presheaf category $\widehat{\mathcal{C}}$ is also regular, because it is complete and cocomplete, with (co)limits computed pointwise. Thus, again, every epi is regular, and epis are stable under pullbacks.

3. ("Fuzzy logic") Let H be a complete Heyting algebra; thus H is a cartesian closed poset with all small joins $\bigvee_i p_i$. The category of *H-presets* has as objects all pairs $(X, e_X : X \to H)$ where X is a set and e_X is a function, called the *existence predicate of X*. For $x \in X$, $e_X(x)$ can be thought of as "the amount by which x exists". A *morphism of presets* is a function $f : X \to Y$ satisfying, for all $x \in X$,

$$e_X(x) \leq e_Y(fx).$$

   This is a regular category, with the following structure.

   - the terminal object is $\top : 1 \to H$,
   - the product of $e_A : A \to H$ and $e_B : B \to H$ is

     $$e_A \wedge e_B : A \times B \to H,$$

     where $(e_a \wedge e_B)(a, b) = e_A(a) \wedge e_B(b)$,
   - the equalizer of two maps $f, g : A \to B$ is their equalizer as functions, $A' = \{a \mid f(a) = g(a)\} \hookrightarrow A$, with the restriction of $e_A : A \to H$ to $A' \subseteq A$.
   - a map $f : A \to B$ is a regular epi if and only if it is a surjective function and for all $b \in B$:

     $$e_B(b) = \bigvee_{f(a)=b} e_A(a)$$

Exercise 3.2.8. Verify that H-presets form a regular category, and compute the regular epi-mono factorization of a map.

The next example deserves to be a proposition.

Proposition 3.2.9. *The category* $\mathsf{Mod}(\mathbb{A}, \mathsf{Set})$ *of set-theoretic models of an algebraic theory* $\mathbb{A}$ *is regular.*

Proof. We sketch a proof, for details see [Bor94, Theorem 3.5.4]. Recall that the objects of $\mathsf{Mod}(\mathbb{A}) = \mathsf{Mod}(\mathbb{A}, \mathsf{Set})$ are $\mathbb{A}$-algebras, which are structures $A = (|A|, f_1, f_2, \ldots)$ where $|A|$ is the underlying set and $f_1, f_2, \ldots$ are the basic operations on $|A|$. Every such $\mathbb{A}$-algebra is also required to satisfy the equational axioms of $\mathbb{A}$. A morphism $h : A \to B$ is a function $h : |A| \to |B|$ that preserves the basic operations.

The category $\mathsf{Mod}(\mathbb{A})$ of $\mathbb{A}$-algebras has small limits, which are created by the forgetful functor $U : \mathsf{Mod}(\mathbb{A}) \to \mathsf{Set}$. Thus the product of $\mathbb{A}$-algebras A and B has as its underlying set $|A \times B| = |A| \times |B|$, and the basic operations of $A \times B$ are computed separately on each factor, and similarly for products of arbitrary (small) families $\prod_i A_i$. An equalizer of morphisms $g, h : A \to B$ has as its underlying set the equalizer of $g, h : |A| \to |B|$, and the basic operations inherited from A.

To see that coequalizers of kernel pairs exist, consider a morphism $h : A \to B$. We can form the quotient $\mathbb{A}$-algebra Q whose underlying set is $|Q| = |A|/{\sim}$, where $\sim$ is the relation defined by

$$x \sim y \iff hx = hy \;,$$

which is just the kernel quotient of the underlying function h. A basic operation $f_Q : |Q|^k \to |Q|$ is induced by the basic operation $f_A : |A|^k \to |A|$ by

$$f_Q\langle [x_1], \ldots, [x_k] \rangle = [f_A\langle x_1, \ldots, x_k \rangle] \;.$$

It is easily verified that this is well-defined, that Q is an $\mathbb{A}$-algebra, and that the canonical quotient map $q : A \to Q$ is the coequalizer of the kernel pair of h.

Lastly regular epis in $\mathsf{Mod}(\mathbb{A})$ are stable because pullbacks and kernel pairs are computed as in Set, and a morphism $h : A \to B$ is a regular epi in $\mathsf{Mod}(\mathbb{A})$ if, and only if, the underlying function $h : |A| \to |B|$ is a regular epi in Set, which is therefore stable under pullback. □

We now know that categories of groups, rings, modules, $\mathcal{C}^\infty$-rings and other algebraic categories are regular. The preceding proposition is useful also for showing that certain structures can*not* be axiomatized by algebraic theories. The category of posets is an example of a category that is not regular; therefore the theory of partial orders cannot be axiomatized solely by equations.

Exercise 3.2.10. Show that Poset is not regular. (Hint: find a regular epi that is not stable under pullback.) Conclude that there is no purely equational reformulation of the cartesian theory of posets.

Exercise* 3.2.11. Is Top regular? Hint: is there is a topological quotient map $q : X \twoheadrightarrow X'$ and a space Y such that $q \times 1_Z : X \times Y \twoheadrightarrow X' \times Y$ is not a quotient map?

Remark 3.2.12 (Exactness)**.** A regular category $\mathcal{C}$ is said to be *exact* [**?**] if *every* equivalence relation (not just those arising as kernel pairs) has a quotient. It can be shown fairly easily that categories of algebras are not just regular but also exact: an equivalence relation in such a category is a congruence relation with respect to the algebraic operations, and its (underlying set) quotient is then necessarily also a homomorphism, and thus a coequalizer of algebras.

Exercise 3.2.13. Prove that the regular epis and monos in a regular category $\mathcal{C}$ form the two classes $(\mathcal{L}, \mathcal{R})$, respectively, of an *orthogonal factorization system* in the following sense:

1. every arrow $f : A \to B$ factors as $f = r \circ l$ with $l \in \mathcal{L}$ and $r \in \mathcal{R}$,

2. $\mathcal{L}$ is the class of all arrows left-orthogonal to all maps in $\mathcal{R}$, and $\mathcal{R}$ is the class of all arrows right-orthogonal to all maps in $\mathcal{L}$, where $l : A \to B$ is said to be *left-orthogonal* to $r : X \to Y$, and r is said to be *right-orthogonal* to l, if for every commutative square as on the outside below,

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "l"'] & X \arrow[d, "r"] \\
B \arrow[r] \arrow[ur, dotted, "d"'] & Y,
\end{tikzcd}$$

there is a unique diagonal arrow d as indicated making both triangles commute.

3.2.2 Images and existential quantifiers

Recall that the poset $\mathsf{Sub}(A)$ is equivalent to the preordered category $\mathsf{Mono}(A)$ of monos into A. If we compose an equivalence functor $\mathsf{Sub}(A) \to \mathsf{Mono}(A)$ with the inclusion $\mathsf{Mono}(A) \to \mathcal{C}/A$ we obtain a (full and faithful) inclusion functor

$$I : \mathsf{Sub}(A) \hookrightarrow \mathcal{C}/A\,. \tag{3.9}$$

In the other direction we have the "image functor" $\mathsf{im} : \mathcal{C}/A \to \mathsf{Sub}(A)$, which maps an object $f : B \to A$ in $\mathcal{C}/A$ to the subobject $\mathsf{im}(f) \rightarrowtail A$.

Exercise 3.2.14. In order to show that im is in fact a functor, prove that $f = g \circ h$ implies $\mathsf{im}(f) \le \mathsf{im}(g)$.

Proposition 3.2.6 says that the image functor is left adjoint to the inclusion functor (3.9),

$$\mathsf{im} \dashv I\ .$$

Furthermore, images are stable in the sense that the following diagram commutes for all $f : A \to B$ (as does the corresponding one with the inclusion I in place of im).

$$\begin{tikzcd}
\mathcal{C}/A \arrow[d, "\mathsf{im}_A"'] & \mathcal{C}/B \arrow[l, "f^*"'] \arrow[d, "\mathsf{im}_B"] \\
\mathsf{Sub}(A) & \mathsf{Sub}(B) \arrow[l, "f^*"]
\end{tikzcd} \tag{3.10}$$

The functor f^* on the top is the "change of base" functor given by pullback of an arbitrary map, and the functor f^* on the bottom is the pullback functor acting on subjects. To see that (3.10) commutes, consider $g : C \to B$ and the following diagram:

$$\begin{tikzcd}
f^*C \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\lrcorner", very near start] \arrow[dd, bend right=60, "f^*g"'] & C \arrow[d, two heads] \arrow[dd, bend left=60, "g"] \\
\cdot \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{im}(g) \arrow[d, tail] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

On the right-hand side we have the factorization of g, which is then pulled back along f. Because monos and regular epis are both stable, this gives a factorization of the pullback f^*g, hence (by the uniqueness of factorizations, Proposition 3.2.4) the claimed equality

$$\mathsf{im}(f^*g) = f^*(\mathsf{im}(g)) \ .$$

Proposition 3.2.15. *A regular category has existential quantifiers. The existential quantifier along $f : A \to B$,*

$$\exists_f : \mathsf{Sub}(A) \longrightarrow \mathsf{Sub}(B),$$

is given by

$$\exists_f[m : M \rightarrowtail A] = \mathsf{im}(f \circ m) \ ,$$

as indicated below.

$$\begin{tikzcd}
M \arrow[r, two heads] \arrow[d, tail, "m"'] & \mathsf{im}(f \circ m) \arrow[d, tail] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

Proof. Recall that composition

$$\Sigma_f : \mathcal{C}/A \longrightarrow \mathcal{C}/B$$

by a map $f : A \to B$ is left adjoint to pullback f^* along f. Thus we are defining $\exists_f = \mathsf{im} \circ \Sigma_f \circ I$ as shown below.

$$\begin{tikzcd}
\mathsf{Sub}(A) \arrow[r, dotted, "\exists_f"] \arrow[d, hook, "I"'] & \mathsf{Sub}(B) \\
\mathcal{C}/A \arrow[r, "\Sigma_f"'] & \mathcal{C}/B \arrow[u, "\mathsf{im}"']
\end{tikzcd}$$

First we verify that $\exists_f \dashv f^*$ on subobjects. For $U \rightarrowtail A$ and $V \rightarrowtail B$:

$$\begin{array}{lr}
\exists_f U \leq V & \text{in } \mathsf{Sub}(B) \\
\hline\hline
\mathsf{im} \circ \Sigma_f \circ I(U) \leq V & \text{in } \mathsf{Sub}(B) \\
\hline\hline
\Sigma_f \circ I(U) \leq I(V) & \text{in } \mathcal{C}/B \\
\hline\hline
I(U) \to f^* I(V) & \text{in } \mathcal{C}/A \\
\hline\hline
I(U) \to I(f^* V) & \text{in } \mathcal{C}/A \\
\hline\hline
U \leq f^* V & \text{in } \mathsf{Sub}(A)
\end{array}$$

In the second step in the above derivation we used the adjunction between $\mathsf{im} : \mathcal{C}/B \to \mathsf{Sub}(B)$ and the inclusion $\mathsf{Sub}(B) \to \mathcal{C}/B$.

The Beck-Chevalley condition follows from stability of image factorizations. Indeed, given a pullback

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

and a subobject $U \rightarrowtail C$, (3.10) gives

$$\begin{aligned}
f^*(\exists_g U) = f^* \circ \mathsf{im} \circ \Sigma_g \circ I(U) = \mathsf{im} \circ f^* \circ \Sigma_g \circ I(U) &= \mathsf{im} \circ \Sigma_k \circ h^* \circ I(U) \\
&= \mathsf{im} \circ \Sigma_k \circ I \circ h^*(U) = \exists_k (h^* U)
\end{aligned}$$

$$\begin{tikzcd}
& f^*U \arrow[r] \arrow[d, tail] \arrow[dl, two heads] \arrow[dr, phantom, "\lrcorner", very near start] & U \arrow[d, tail] \arrow[dr, two heads] & \\
\exists_k f^*U \arrow[dr, tail] & D \arrow[r, "h"] \arrow[d, "k"'] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "g"] & \exists_g U \arrow[dl, tail] \\
& A \arrow[r, "f"'] & B &
\end{tikzcd}$$

as required. □

Summarizing the results of this section, we have the following.

Proposition 3.2.16. *In any regular category, for every map $f : A \to B$ we have the following situation, where f^* is pullback:*

$$\begin{tikzcd}
\mathsf{Sub}(A) \arrow[r, shift right, "\exists_f"'] \arrow[d, hook, shift left, "I"] & \mathsf{Sub}(B) \arrow[l, shift right, "f^*"'] \arrow[d, hook, shift left, "I"] \\
\mathcal{C}/A \arrow[u, shift left, "\mathsf{im}"] \arrow[r, shift right, "\Sigma_f"'] & \mathcal{C}/B \arrow[l, shift right, "f^*"'] \arrow[u, shift left, "\mathsf{im}"]
\end{tikzcd}$$

with adjunctions

$$\exists_f \dashv f^*, \quad \mathsf{im} \dashv I, \quad \Sigma_f \dashv f^*$$

and natural isos

$$f^* \circ \mathsf{im} \cong \mathsf{im} \circ f^*, \quad f^* \circ I \cong I \circ f^*.$$

Note, moreover, that

$$\exists_f \circ \mathsf{im} \cong \mathsf{im} \circ \Sigma_f$$

then follows.

Finally, we call attention to the following special fact.

Proposition 3.2.17 (Frobenius Reciprocity)**.** *Given a map $f : A \to B$ and subobjects $U \leq A$ and $V \leq B$, the following equation holds in $\mathsf{Sub}(B)$.*

$$\exists_f(U \wedge f^*V) = \exists_f U \wedge V$$

Exercise 3.2.18. Prove Frobenius reciprocity, using the following diagram.

$$\begin{tikzcd}
 & U \wedge f^*V \arrow[rr] \arrow[d] \arrow[dl] & & \exists_f U \wedge V \arrow[d] \arrow[dl] \\
 & f^*V \arrow[rr] \arrow[ddl] & & V \arrow[ddl, tail] \\
U \arrow[rr] \arrow[d, tail] & & \exists_f U \arrow[d] & \\
A \arrow[rr, "f"'] & & B &
\end{tikzcd}$$

3.2.3 Regular theories

A regular category has finite limits and image factorizations, therefore it allows us to interpret a type theory with the terminal type and binary products, and a logic with equality, conjunction, and existential quantifiers. This system is called *regular logic.*

Definition 3.2.19. A (many-sorted) *regular theory* $\mathbb{T}$ is a (many-sorted) type theory together with a set of axioms expressed in the fragment of logic built from $=$, $\top$, $\wedge$, and $\exists$.

In more detail, a regular theory consists of the following data, extending the notion of cartesian theory from section **??**.

- basic type symbols $A_1, \ldots, A_k$,
- basic function symbols $f, \ldots$ (with signature) $(A_1, \cdots, A_m; B)$,
- basic relation symbols $R, \ldots$ (with signature) $(A_1, \cdots, A_n)$.

We then define by induction the set of terms in context,

$$\Gamma \mid t : A\,,$$

as well as the formulas in context,

$$\Gamma \mid \varphi \ \texttt{pred}\,.$$

Here is the first place where things differ from cartesian logic; we extend the formation rules for cartesian formulas (section 3.1.3) by the further clause:

6. Existential Quantifier:

$$\frac{\Gamma, x : A \mid \varphi \ \texttt{pred}}{\Gamma \mid \exists x : A.\, \varphi \ \texttt{pred}}$$

(We also add the evident additional clause for sustitution of terms into existentially quantified formulas, namely $(\exists x : A.\, \varphi)[t/y] = \exists x : A.\, (\varphi[t/y])$.) This defines the notion of a *regular formula*, i.e. ones built from the atomic formulas $s = t$ and $R(t_1, \ldots, t_n)$ using the logical operations $\top$, $\wedge$, and $\exists$.

A regular theory then includes, finally, a set of axioms of the form

$$\Gamma \mid \varphi \vdash \psi$$

where φ, ψ are *regular formulas.*

Example 3.2.20. 1. A ring A (with unit 1) is called *von Neumann regular* if for every element a there is at least one element x for which $a = a \cdot x \cdot a$. Such an x may be thought of as a "weak inverse" of a. The theory of *von Neumann regular rings* is thus an extension of the usual theory of rings with unit by adding the single axiom

$$a : A \mid \top \vdash \exists\, x : A\,.\, a = a \cdot x \cdot a$$

2. A perhaps more familiar example is the theory of categories, with two basic types A, O for arrows and objects, 3 basic function symbols $\mathrm{dom}, \mathrm{cod} : (A; O)$ and $\mathrm{id} : (O; A)$ and one basic relation symbol $C : (A, A, A)$, where the latter is for the relation $C(x, y, z) =$ "z is the composite of x and y". The axioms for C are as follows (with abbreviated notation for the context):

$$\begin{aligned} x, y, z : A \mid C(x, y, z) &\vdash \mathrm{cod}(x) = \mathrm{dom}(y) \wedge \mathrm{dom}(z) = \mathrm{dom}(x) \wedge \mathrm{cod}(z) = \mathrm{cod}(y) \\ x, y : A \mid \mathrm{cod}(x) = \mathrm{dom}(y) &\vdash \exists z.\, C(x, y, z) \\ x, y, z, z' : A \mid C(x, y, z) \wedge C(x, y, z') &\vdash z = z' \end{aligned}$$

Recall the previous versions of the theory of categories as cartesian theories in 3.1.23. Are the homomorphisms of categories, as models of a regular theory, the same thing as functors?

3. The theory of an *inhabited object* has a single type A, no function or relation symbols, and the single axiom:

$$\cdot \mid \top \vdash \exists x : A.\, x = x$$

A model is an object that is "inhabited" by at least one (unnamed) element, but the homomorphisms need not preserve anything – in this sense being inhabited is a *property*, not a *structure.*

The *rules of inference* of regular logic are those of cartesian logic (section 3.1.3), with an additional rule for the existential quantifier:

8. Existential Quantifier:

$$\frac{y : B, x : A \mid \varphi \vdash \vartheta}{\overline{y : B \mid \exists\, x : A\,.\, \varphi \vdash \vartheta}}$$

Note that the lower judgement is well-formed only if $x : A$ does not occur freely in ϑ.

We also add a rule coresponding to Frobenius reciprocity, Proposition 3.2.17, in the form

9. Frobenius:
$$x : A \mid (\exists y : B.\varphi) \wedge \psi \ \vdash \ \exists y : B.(\varphi \wedge \psi)$$
provided the variable $y : B$ does not occur freely in ψ.

Note that the converse of Frobenius is easily derivable, so we have the interderivability of $(\exists y : B.\varphi) \wedge \psi$ and $\exists y : B.(\varphi \wedge \psi)$ when $y : B$ is not free in ψ. The Frobenius rule will be derivable in the extended system of Heyting logic (see Proposition 3.3.15), and could be made derivable in a suitably formulated system of regular logic using multi-sequents $\Gamma \mid \varphi_1, \ldots, \varphi_n \vdash \psi$.

Semantics of regular theories

Turning to semantics, an *interpretation* of a regular theory $\mathbb{T}$ in a regular category $\mathcal{C}$ extends the notion for cartesian logic (section 3.1.3), and is given by the following data:

1. Each basic sort A is interpreted as an object $[\![A]\!]$.

2. Each basic constant f with signature $(A_1, \ldots, A_n; B)$ is interpreted as a morphism $[\![f]\!] : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![B]\!]$.

3. Each basic relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted as a subobject $[\![R]\!] \in \mathsf{Sub}([\![A_1]\!] \times \cdots \times [\![A_1]\!])$.

This is the same as for cartesian logic, as is the extension of the interpretation to all terms,

$$[\![\Gamma \mid t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$$

For the formulas, we extended the interpretation to cartesian formulas as before (section **??**),

$$[\![\Gamma \mid \varphi]\!] \rightarrowtail [\![\Gamma]\!] \,.$$

Finally, existential formulas $\exists\, x : A \,.\, \varphi$ are interpreted by the existential quantifiers in the regular category,

$$[\![\Gamma \mid \exists\, x : A \,.\, \varphi]\!] = \exists_A [\![\Gamma, x : A \bigm| \varphi]\!] \,,$$

where

$$\exists_A = \exists_\pi : \mathsf{Sub}([\![\Gamma]\!] \times [\![A]\!]) \to \mathsf{Sub}([\![\Gamma]\!])$$

is the existential quantifier along the projection $\pi : [\![\Gamma]\!] \times [\![A]\!] \to [\![\Gamma]\!]$.

The following is immediate from these definitions, and the considerations in section **??**.

Proposition 3.2.21. *The rules of regular logic are sound with respect to the interpretation in regular categories.*

Exercise 3.2.22. Prove this.

If all the axioms of $\mathbb{T}$ hold in a given interpretation, then we again say that the interpretation is a *model* of the theory $\mathbb{T}$. Morphisms of models are just morphisms of the underlying cartesian structures. Thus for any regular theory $\mathbb{T}$ and regular category $\mathcal{C}$, there is a *category of models*,

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

Moreover, this semantic category is functorial in $\mathcal{C}$ with respect to regular functors $\mathcal{C} \to \mathcal{D}$, which, recall, preserve finite limits and regular epis. Indeed, if $F : \mathcal{C} \to \mathcal{D}$ is regular then given a model M in $\mathcal{C}$ with underlying cartesian structure $[\![A]\!]_M, [\![f]\!]_M, [\![R]\!]_M$, etc., we can determine an interpretation FM in $\mathcal{D}$ by setting:

$$[\![A]\!]_{FM} = F([\![A]\!]_M),\ [\![f]\!]_{FM} = F([\![f]\!]_M),\ [\![R]\!]_{FM} = F([\![f]\!]_M)$$

etc., and these will have the correct types (up to isomorphism). To show that FM is a $\mathbb{T}$-model, if M is one and F is regular, consider an axiom of $\mathbb{T}$ of the form $\Gamma \mid \varphi \vdash \psi$. Satisfaction by M means that $[\![\Gamma \mid \varphi]\!]_M \leq [\![\Gamma \mid \psi]\!]_M$ in $\mathsf{Sub}([\![\Gamma]\!]_M)$, which in turn means that there is a (necessarily unique) factorization,

$$\begin{tikzcd}
{[\![\Gamma \mid \varphi]\!]_M} \arrow[rr, dotted] \arrow[rd, tail] & & {[\![\Gamma \mid \psi]\!]_M} \arrow[ld, tail] \\
& {[\![\Gamma]\!]_M} & ,
\end{tikzcd}$$

Applying the cartesian functor F will result in an inclusion of subobjects $F[\![\Gamma \mid \varphi]\!]_M \leq F[\![\Gamma \mid \psi]\!]_M$ in $\mathsf{Sub}(F[\![\Gamma]\!]_M) = \mathsf{Sub}([\![\Gamma]\!]_{FM})$. Thus is clearly suffices to show that for any regular formula φ,

$$F[\![\Gamma \mid \varphi]\!]_M = [\![\Gamma \mid \varphi]\!]_{FM} .$$

This is an easy induction on φ, using the regularity of F.

Proposition 3.2.23. *Given a regular functor $F : \mathcal{C} \to \mathcal{D}$, taking images determines a functor*

$$F_* : \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{D}) .$$

Proof. It only remains show the effect of F_* on morphisms of models. But these are just homomorphisms of the underlying cartesian structure, so they are clearly preserved by the cartesian functor F. □

An associated result, which we will need, is the following.

Proposition 3.2.24. *Given regular categories $\mathcal{C}$ and $\mathcal{D}$ and a model M in $\mathcal{C}$, evaluation at M determines a functor*

$$\mathsf{eval}_M : \mathsf{Reg}(\mathcal{C}, \mathcal{D}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{D}) ,$$

which is natural in $\mathcal{D}$.

The proof is straightforward and can be left as an exercise. The naturality means that for any a regular functor $G : \mathcal{D} \longrightarrow \mathcal{D}'$, the following commutes (up to natural isomorphism, as usual):

$$\begin{tikzcd}
\mathsf{Reg}(\mathcal{C}, \mathcal{D}) \arrow[r, "\mathsf{eval}_M"] \arrow[d, "\mathsf{Reg}(\mathcal{C}, G)"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{D}) \arrow[d, "G_*"] \\
\mathsf{Reg}(\mathcal{C}, \mathcal{D}') \arrow[r, "\mathsf{eval}_M"'] & \mathsf{Mod}(\mathbb{T}, \mathcal{D}')
\end{tikzcd}$$

Exercise 3.2.25. Prove this.

Exercise 3.2.26. Show that for any small category $\mathbb{C}$ and regular theory $\mathbb{T}$, there is an equivalence between models in the functor category and functors into the category of models,

$$\mathsf{Mod}(\mathbb{T}, \mathsf{Set}^{\mathbb{C}}) \;\simeq\; \mathsf{Mod}(\mathbb{T})^{\mathbb{C}} .$$

Hint: this is just as for the algebraic and cartesian cases.

3.2.4 The classifying category of a regular theory

We will next show that the framework of *functorial semantics* applies to regular logic and regular categories: there is a *classifying category* $\mathcal{C}_{\mathbb{T}}$ for $\mathbb{T}$-models, for which there is an equivalence, natural in $\mathcal{C}$,

$$\mathsf{Reg}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \;\simeq\; \mathsf{Mod}(\mathbb{T}, \mathcal{C}) ,$$

where $\mathsf{Reg}(-,-)$ is the category of regular functors and natural transformations.

Remark 3.2.27. The construction of $\mathcal{C}_{\mathbb{T}}$, and the corollary completeness theorem, are analogous to the way of proving the completeness theorem for (say, classical) propositional logic that we used in Chapter 2: one first constructs the *Lindenbaum-Tarski algebra* of propositional logic with respect to a propositional theory $\mathbb{T}$ (a set of formulas) as the set $\mathsf{PL} = \{\varphi \mid \varphi \text{ a propositional formula}\}$, quotiented by $\mathbb{T}$-provable logical equivalence, $\varphi \sim_{\mathbb{T}} \psi$ iff $\mathbb{T} \vdash \varphi \leftrightarrow \psi$,

$$\mathcal{B}_{\mathbb{T}} = \mathsf{PL}/\sim_{\mathbb{T}} \ .$$

The quotient set $\mathcal{B}_{\mathbb{T}}$ becomes a Boolean algebra by defining the Boolean operations in terms of the expected propositional logical analogues,

$$[\varphi] \wedge [\psi] = [\varphi \wedge \psi] , \quad \neg[\varphi] = [\neg\varphi] , \quad [\top] = 1 , \quad \text{etc.} .$$

One then has a Boolean-valuation of PL in $\mathcal{B}_{\mathbb{T}}$, namely $[-]$, for which

$$[\varphi] = [\psi] \quad \text{iff} \quad \mathbb{T} \vdash \varphi \leftrightarrow \psi .$$

In particular, we have $[\varphi] = 1$ in $\mathcal{B}_{\mathbb{T}}$ iff $\mathbb{T} \vdash \varphi$. Classical completeness with respect to valuations in the Boolean algebra $\mathbf{2} = \{1, 0\}$ then follows e.g. from Stone's representation theorem, which embeds the Boolean algebra $\mathcal{B}_{\mathbb{T}}$ into a powerset $\mathcal{P}(X) \cong \mathbf{2}^X$, where X is the set of prime ideals in $\mathcal{B}_{\mathbb{T}}$, corresponding to Boolean homomorphisms $\mathcal{B}_{\mathbb{T}} \to \mathbf{2}$, which in turn correspond to Boolean valuations of the language PL, i.e. "rows of a truth table".

Our syntactic construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ can be regarded as a generalization of this method, with $\mathcal{C}_{\mathbb{T}}$ as the "Lindenbaum-Tarski category" of the (regular) theory $\mathbb{T}$. This will give a completeness theorem with respect to models in regular categories, which can in turn be specialized to Set-valued completeness by embedding $\mathcal{C}_{\mathbb{T}}$ into a "power of Set", i.e. Set^X for a set X. The elements of X will be regular functors $\mathcal{C}_{\mathbb{T}} \to \mathsf{Set}$, corrresponding to "classical" models of $\mathbb{T}$ in Set. See Section 3.2.6 below for the second step.

We first sketch the construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ of an arbitrary regular theory $\mathbb{T}$ (a more detailed account can be found in [But98, Joh03]). An object of $\mathcal{C}_{\mathbb{T}}$ is represented by a formula in context,

$$[\Gamma \mid \varphi],$$

where $\Gamma \mid \varphi$ `pred`. Two such objects $[\Gamma \mid \varphi]$ and $[\Gamma \mid \psi]$ are equal if $\mathbb{T}$ proves both

$$\Gamma \mid \varphi \vdash \psi \; , \qquad\qquad \Gamma \mid \psi \vdash \varphi \; .$$

Objects which differ only in the names of free variables are also considered equal:

$$[x : A \mid \varphi] = [y : A \mid \varphi[y/x]] \qquad \text{(no } y \text{ in } \varphi)$$

A morphism

$$[x : A \mid \varphi] \xrightarrow{\rho} [y : B \mid \psi]$$

is represented by a formula $x : A, y : B \mid \rho$ such that $\mathbb{T}$ proves that ρ is a *functional relation* from φ to ψ:

$$\begin{aligned} x : A \mid \varphi &\vdash \exists y : B \, . \, \rho && \text{(total)} \\ x : A, y : B, z : B \mid \rho \wedge \rho[z/y] &\vdash y = z && \text{(single-valued)} \\ x : A, y : B \mid \rho &\vdash \varphi \wedge \psi && \text{(well-typed)} \end{aligned}$$

Two functional relations ρ and σ represent the same morphism if $\mathbb{T}$ proves both

$$x : A, y : B \mid \rho \vdash \sigma \; , \qquad\qquad x : A, y : B \mid \sigma \vdash \rho \; .$$

Relations which only differ in the names of free variables are also considered equal.

(Strictly speaking, a morphism

$$[x : A, y : B \mid \rho] : [x : A \mid \varphi] \to [y : B \mid \psi]$$

should be taken to be the triple

$$\big([x : A, y : B \mid \rho], \; [x : A \mid \varphi], \; [y : B \mid \psi]\big)$$

so that one knows what the domain and codomain are, but we shall often write simply

$$\rho : [x : A \mid \varphi] \to [y : B \mid \psi]$$

since the rest can be recovered from that much data.)

The identity morphism on $[x : A \mid \varphi]$ is

$$1_{[x:A|\varphi]} = [x : A, x' : A \mid (x = x') \wedge \varphi] : [x : A \mid \varphi] \to [x' : A \mid \varphi[x'/x]] \ .$$

Note that we used the variable substitution $\varphi[x'/x]$ and the identification $[x : A \mid \varphi] = [x' : A \mid \varphi[x'/x]]$ in order to make this definition.

Composition of morphisms

$$\begin{tikzcd} {[x : A \mid \varphi]} \arrow[r, "\rho"] & {[y : B \mid \psi]} \arrow[r, "\tau"] & {[z : C \mid \theta]} \end{tikzcd}$$

is given by the relational product,

$$\tau \circ \rho = (\exists\, y : B \,.\, (\rho \wedge \tau)) \ .$$

Of course, one needs to check that this *is* a morphism from φ to ϑ, i.e. that it is total, single-valued, and well-typed. We leave the detailed proof that $\mathcal{C}_{\mathbb{T}}$ is a category as an exercise; let us just show how to prove that composition of morphisms is associative. Given morphisms

$$\begin{tikzcd} {[x : A \mid \varphi]} \arrow[r, "\rho"] & {[y : B \mid \psi]} \arrow[r, "\tau"] & {[z : C \mid \theta]} \arrow[r, "\sigma"] & {[u : D \mid \zeta]} \end{tikzcd}$$

we need to derive in context $x : A, u : D$

$$\exists\, z : C \,.\, ((\exists\, y : B \,.\, (\rho \wedge \tau)) \wedge \sigma) \ \dashv\vdash\ \exists\, y : B \,.\, (\rho \wedge (\exists\, z : C \,.\, (\tau \wedge \sigma)))$$

This follows easily with repeated application of the Frobenius rule (Section 3.2.3).

Exercise 3.2.28. Extend the definition of $\mathcal{C}_{\mathbb{T}}$ to morphisms between objects with arbitrary contexts,

$$\begin{tikzcd} {[\Gamma \mid \varphi]} \arrow[r, "\rho"] & {[\Delta \mid \psi]} \end{tikzcd}$$

(use relations $\Gamma, \Delta \mid \rho$), and provide a proof that $\mathcal{C}_{\mathbb{T}}$ is a category.

Proposition 3.2.29. *The category $\mathcal{C}_{\mathbb{T}}$ is regular.*

Proof. We sketch the constructions required for regularity.

- The terminal object is $[\cdot \mid \top]$.
- The product of $[x : A \mid \varphi]$ and $[y : B \mid \psi]$, where x and y are distinct variables, is the object
$$[x : A, y : B \mid \varphi \wedge \psi] \ .$$

The first projection from the product is

$$x : A, y : B, x' : A \mid x = x' \land \varphi \land \psi\,,$$

and the second projection is

$$x : A, y : B, y' : B \mid y = y' \land \varphi \land \psi\,,$$

where we rename the codomains of the projections $[x : A \mid \varphi] = [x' : A \mid \varphi[x'/x]]$, etc., to make the context variables distinct.

- An equalizer of morphisms

$$\begin{tikzcd}[x : A \mid \varphi] \arrow[r, shift left, "\rho"] \arrow[r, shift right, "\tau"'] & {[y : B \mid \psi]}\end{tikzcd}$$

  is

$$\begin{tikzcd}[x : A \mid \exists\, y : B\,.\,(\rho \land \tau)] \arrow[r, "\varepsilon"] & {[x' : A \mid \varphi[x'/x]]}\end{tikzcd}$$

  where ε is the morphism

$$x : A, x' : A \mid (x = x') \land \exists\, y : B\,.\,(\rho \land \tau)\ .$$

- Finally, let us consider coequalizers of kernel pairs. The kernel pair of a map

$$\rho : [x : A \mid \varphi] \longrightarrow [y : B \mid \psi]$$

  is

$$\begin{tikzcd}K \arrow[r, shift left, "\kappa_1"] \arrow[r, shift right, "\kappa_2"'] & {[x : A \mid \varphi]}\end{tikzcd}$$

  where K is the object

$$[u : A, v : A \mid \exists\, y : B\,.\,(\rho[u/x] \land \rho[v/x])]\ ,$$

  the morphism κ_1 is

$$u : A, v : A, x : A \mid (u = x) \land \exists\, y : B\,.\,(\rho[u/x] \land \rho[v/x])\ ,$$

  and κ_2 is

$$u : A, v : A, x : A \mid (v = x) \land \exists\, y : B\,.\,(\rho[u/x] \land \rho[v/x])\ .$$

  Now the coequalizer of κ_1 and κ_2 can be shown to be the morphism

$$\begin{tikzcd}[x : A \mid \varphi] \arrow[r, "\rho"] & {[y : B \mid \exists x : A.\, \rho]}\end{tikzcd}\ ,$$

  where $[y : B \mid \exists\, x : A\,.\,\rho]$ is the image of ρ, as a subobject of $[y : B \mid \psi]$.

The following lemma shows that regular epis are stable under pullback. □

Lemma 3.2.30. *1. A map* $\rho : [x : A \mid \varphi] \longrightarrow [y : B \mid \psi]$ *is a regular epi if and only if*

$$y : B \mid \psi \vdash \exists x : A.\, \rho$$

2. Regular epis are stable under pullback in $\mathcal{C}_{\mathbb{T}}$*.*

Proof. For (1), suppose $\rho : [x : A \mid \varphi] \to [y : B \mid \psi]$ is a regular epi. We claim first that if ρ factors through some subobject $U \rightarrowtail [y : B \mid \psi]$ then $U = [y : B \mid \psi]$ is the maximal suboject. Indeed, since ρ is regular epi it is a coequalizer of its kernel pair. But if ρ factors through a subobject $U \rightarrowtail [y : B \mid \psi]$, say by $r : [x : A \mid \varphi] \to U$, then r is also a coequalizer of the kernel pair of ρ, as one can easily check. Thus $U \rightarrowtail [y : B \mid \psi]$ must be iso.

Now, up to iso, every $U \rightarrowtail [y : B \mid \psi]$ is of the form $U = [y : B \mid \vartheta]$ with $y \mid \vartheta \vdash \psi$, and ρ factors through $[y : B \mid \vartheta]$ iff

$$y : B \mid \exists x : A\,.\rho \vdash \vartheta\,.$$

Thus for all ϑ we have that:

$$(y : B \mid \exists x : A\,.\rho \vdash \vartheta) \;\Rightarrow\; (y : B \mid \psi \vdash \vartheta)\,.$$

Whence $y : B \mid \psi \vdash \exists x : A\,.\rho$. The convere is immediate from the specification of the kernel quotient above.

For (2), suppose we have a pullback diagram, which has the form indicated below.

$$\begin{tikzcd}
{[x : A, y : B \mid \varphi \wedge \psi \wedge \exists z : C.\,(\sigma \wedge \rho)]} \arrow[r, "\rho^*\sigma"] \arrow[d, "\sigma^*\rho"'] & {[y : B \mid \psi]} \arrow[d, "\rho"] \\
{[x : A \mid \varphi]} \arrow[r, "\sigma"'] & {[z : C \mid \vartheta]}
\end{tikzcd}$$

The maps $\sigma^*\rho$ and $\rho^*\sigma$ are represented by the relations:

$$\begin{aligned}
\sigma^*\rho &= \big(x : A,\, y : B,\, x' : A \mid x = x' \wedge \varphi \wedge \psi \wedge \exists z : C.\,(\sigma \wedge \rho)\big) \\
\rho^*\sigma &= \big(x : A,\, y : B,\, y' : B \mid y = y' \wedge \varphi \wedge \psi \wedge \exists z : C.\,(\sigma \wedge \rho)\big)
\end{aligned}$$

If ρ is regular epi, then by (1) we have

$$z : C \mid \vartheta \vdash \exists y : B.\, \rho\,. \tag{3.11}$$

To show that the pullback $\sigma^*\rho$ is regular epi, again by (1) we need to show

$$x' : A \mid \varphi[x'/x] \vdash \exists x : A\, \exists y : B.\, \big(x = x' \wedge \varphi \wedge \psi \wedge \exists z : C.\,(\sigma \wedge \rho)\big)\,. \tag{3.12}$$

We can make use thereby of the functionality of σ and ρ, specifically we have

$$x : A,\, z : C \mid \sigma \vdash \varphi \wedge \vartheta \qquad \text{and} \qquad x : A \mid \varphi \vdash \exists z : C.\, \sigma\,. \tag{3.13}$$

The result now follows by a simple deduction. □

Exercise 3.2.31. Show that in $\mathcal{C}_{\mathbb{T}}$ the regular-epi mono factorization of a morphism $\rho : [x : A \mid \varphi] \to [y : B \mid \psi]$ is given by

$$[x : A \mid \varphi] \xrightarrow{\rho} [y : B \mid \exists\, x : A \,.\, \rho] \xrightarrow{\iota} [z : B \mid \psi[z/y]]$$

where ι is the morphism

$$y : B, z : B \mid (y = z) \land (\exists\, x : A \,.\, \rho) \;.$$

Theorem 3.2.32 (Functorial semantics for regular logic)**.** *For any regular theory $\mathbb{T}$, the syntactic category $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models in regular categories. Specifically, for any regular category $\mathcal{C}$, there is an equivalence of categories*

$$\mathsf{Reg}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C}) \tag{3.14}$$

which is natural in $\mathcal{C}$. In particular, there is a universal model U in $\mathcal{C}_{\mathbb{T}}$.

Proof. We have just constructed $\mathcal{C}_{\mathbb{T}}$ and shown that it is regular.

The universal model U, corresponding to the identity functor $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}}$ under (3.14), is determined as follows:

- Each sort A is interpreted by the object $[x : A \mid \top]$
- A basic constant f with signature $(A_1, \ldots, A_n; B)$ is interpreted by the formula

  $$x_1 : A_1, \ldots, x_n : A_n, y : B \mid f(x_1, \ldots, x_n) = y \;.$$

  which is plainly a functional relation and thus a morphism $[\![A_1]\!] \times \cdots \times [\![A_n]\!] \longrightarrow [\![B]\!]$.
- A relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted by the subobject represented by the morphism

  $$\rho : [x_1 : A_1, \ldots, x_n : A_n \mid R(x_1, \ldots, x_n)] \longrightarrow [y_1 : A_1, \ldots, y_n : A_n \mid \top]$$

  where ρ is the formula

  $$x_1 : A_1, \ldots, x_n : A_n, y_1 : A_1, \ldots, y_n : A_n \mid R(x_1, \ldots, x_n) \land x_1 = y_1 \land \cdots \land x_n = y_n \;,$$

  which is easily shown to be monic.

It is now straightforward to show that with respect to this structure, a formula $\Gamma \mid \varphi$ is interpreted as (the subobject determined by) the map

$$\iota : [\Gamma \mid \varphi] \longrightarrow [\Gamma \mid \top]$$

where ι is the formula

$$\Gamma, \Gamma' \mid \Gamma = \Gamma' \land \varphi \;,$$

(with obvious abbreviations) which, again, is easily shown to be monic. Moreover, for any formulas $\Gamma \mid \varphi$ and $\Gamma \mid \psi$ we then have

$$U \models \Gamma \mid \varphi \vdash \psi \quad \iff \quad \mathbb{T} \text{ proves } \Gamma \mid \varphi \vdash \psi \,.$$

Thus in particular U is indeed a $\mathbb{T}$-model.

We next construct a functor $\mathsf{Reg}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \to \mathsf{Mod}(\mathbb{T}, \mathcal{C})$. Suppose $\mathcal{C}$ is regular and $F : \mathcal{C}_{\mathbb{T}} \longrightarrow \mathcal{C}$ a regular functor, then by Proposition 3.2.24, applying F to U determines a model FU in $\mathcal{C}$ with

$$[\![A]\!]_{FU} = F([\![A]\!]_U)\,,$$

and similarly for the other parts of the structure f, R, etc. Satisfaction of an entailment $\Gamma \mid \varphi \vdash \psi$ is preserved, because the interpretation of the logical operations is determined by the regular structure: pullbacks, images, etc., so that $[\![\varphi]\!]_U \leq [\![\psi]\!]_U$ in $\mathsf{Sub}([\![\Gamma]\!])$ implies

$$[\![\varphi]\!]_{FU} = F([\![\varphi]\!]_U) \leq F([\![\psi]\!]_U) = [\![\psi]\!]_{FU}$$

in $\mathsf{Sub}([\![\Gamma]\!]_{FU})$.

Moreover, just as for algebraic structures, every natural transformation between regular functors $\vartheta : F \Rightarrow G$ determines a homomorphism of the evaluated models by taking components $\vartheta_U : FU \to GU$. In this way, as in Proposition 3.2.24, evaluation at U is a functor

$$\mathsf{eval}_U : \mathsf{Reg}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \longrightarrow \mathsf{Mod}(\mathbb{T}, \mathcal{C})\,.$$

We claim that this functor, which is the one mentioned in (3.14), is full and faithful and essentially surjective. The naturality in $\mathcal{C}$ of the equivalence then follows directly from its determination by evaluation at U and Proposition 3.2.24.

To see that eval_U is essentially surjective, let M be a model in $\mathcal{C}$. We will define a regular functor

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

with $M^\sharp(U) \cong M$. Since M is a model, there are objects $[\![A]\!]_M$ interpreting each type A, as well as interpretations

$$[\![\Gamma \mid \varphi]\!] \rightarrowtail [\![\Gamma]\!]$$

for all formulas and

$$[\![\Gamma \mid t : B]\!] : [\![\Gamma]\!] \longrightarrow [\![B]\!]$$

for all terms. Using these, we determine the functor $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ by taking an object $[\Gamma \mid \varphi]$ to $[\![\Gamma \mid \varphi]\!]_M$, i.e. the domain of a mono representing the subobject $[\![\Gamma \mid \varphi]\!]_M \rightarrowtail [\![\Gamma]\!]_M$. Thus, for the record,

$$M^\sharp[\Gamma \mid \varphi] \;=\; [\![\Gamma \mid \varphi]\!]_M\,.$$

In the verification that those formulas in context $[\Gamma \mid \varphi]$ that are identified in $\mathcal{C}_{\mathbb{T}}$ are also identified in $\mathcal{C}$, we use the fact that the rules of inference for regular logic are sound in the regular category $\mathcal{C}$. Note in particular that for each basic type A, we then have

$$M^\sharp([\![A]\!]_U) = M^\sharp([x : A \mid \top]) \cong [\![x : A \mid \top]\!]_M \cong [\![A]\!]_M\,,$$

so that $M^\sharp(U) \cong M$ as required.

Functional relations in $\mathcal{C}_\mathbb{T}$ determine functional relations in $\mathcal{C}$, again by soundness, which determines the action of $M^\sharp$ on arrows, as well as the functoriality of these assignments.

Finally, to show that eval_U is full and faithful, let $F, G : \mathcal{C}_\mathbb{T} \longrightarrow \mathcal{C}$ be regular functors classifying models FU and GU, and let $h : FU \to GU$ be a model homomorphism. We then have maps

$$h_{[x:A\mid\top]} : F([x : A \mid \top]) \longrightarrow G([x : A \mid \top])$$

for all basic types A, and these commute with the interpretations of the function symbols f, and preserve the basic relations R, in the obvious sense, because h is a homomorphism. It only remains to determine the components

$$h_{[\Gamma\mid\varphi]} : F([\Gamma \mid \varphi]) \to G([\Gamma \mid \varphi])\,, \tag{3.15}$$

and to show that they commute with all maps $\rho : [\Gamma \mid \varphi] \to [\Delta \mid \psi]$. Define

$$h_{[\Gamma\mid\varphi]} : F[\Gamma \mid \varphi] = [\![\Gamma \mid \varphi]\!]_{FU} \longrightarrow [\![\Gamma \mid \varphi]\!]_{GU} = G[\Gamma \mid \varphi]$$

by induction on the structure of φ. The base cases involving the primitive relations $R, ...$ and equality of terms are given by the assumption that $h : FU \to GU$ is a model homomorphism, so we just need to check that for every definable subobject

$$[\![\Gamma \mid \varphi]\!]_{FU} \rightarrowtail [\![\Gamma \mid \top]\!]_{FU}$$

the following diagram can be filled in as indicated.

$$\begin{tikzcd}
{[\![\Gamma \mid \varphi]\!]_{FU}} \arrow[r, tail] \arrow[d, dotted, "h_{[\Gamma\mid\varphi]}"'] & {[\![\Gamma \mid \top]\!]_{FU}} \arrow[d, "h_{[\Gamma\mid\top]}"] \\
{[\![\Gamma \mid \varphi]\!]_{GU}} \arrow[r, tail] & {[\![\Gamma \mid \top]\!]_{GU}}
\end{tikzcd} \tag{3.16}$$

Suppose we have e.g. $\varphi = \exists x : A.\, \psi$, and we have already determined

$$h_{[\Gamma, x:A\mid\psi]} : [\![\Gamma, x : A \mid \psi]\!]_{FU} \longrightarrow [\![\Gamma, x : A \mid \psi]\!]_{GU}\,.$$

An easy diagram chase shows that there is a unique $h_{[\Gamma\mid\exists x:A.\,\psi]}$ determined by the image factorizations indicated below.

$$\begin{tikzcd}
{[\![\Gamma, x : A \mid \psi]\!]_{FU}} \arrow[r, two heads] \arrow[d, "h_{[\Gamma,x:A\mid\psi]}"'] & {[\![\Gamma \mid \varphi]\!]_{FU}} \arrow[r, tail] \arrow[d, dotted, "h_{[\Gamma\mid\exists x:A.\,\psi]}"'] & {[\![\Gamma \mid \top]\!]_{FU}} \arrow[d, "h_{[\Gamma\mid\top]}"] \\
{[\![\Gamma, x : A \mid \psi]\!]_{GU}} \arrow[r, two heads] & {[\![\Gamma \mid \varphi]\!]_{GU}} \arrow[r, tail] & {[\![\Gamma \mid \top]\!]_{GU}}
\end{tikzcd}$$

The other cases are even more direct. Thus we have defined the components (3.15); we leave the required naturality with respect to all maps $\rho : [\Gamma \mid \varphi] \to [\Delta \mid \psi]$ as an exercise. □

Exercise 3.2.33. Prove the naturality of the maps (3.15), using the following trick. In any category with finite products, suppose we have objects and arrows

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[d, "\beta"] \\ C \arrow[r, "g"'] & D \end{tikzcd} \tag{3.17}$$

Let $\hat{f} = \langle 1_A, f \rangle : A \rightarrowtail A \times B$ be the graph of f, and similarly for $\hat{g} : C \rightarrowtail C \times D$. Then the diagram (3.17) commutes iff the following one does.

$$\begin{tikzcd} A \arrow[r, tail, "\hat{f}"] \arrow[d, "\alpha"'] & A \times B \arrow[d, "\alpha \times \beta"] \\ C \arrow[r, tail, "\hat{g}"'] & C \times D \end{tikzcd}$$

Corollary 3.2.34. *The rules of regular logic are sound and complete with respect to semantics in regular categories: a regular theory* $\mathbb{T}$ *proves an entailment*

$$\Gamma \mid \varphi \vdash \psi \tag{3.18}$$

if, and only if, every model of $\mathbb{T}$ *satisfies it.*

Proof. As for algebraic logic, soundness follows from classification (although we have of course already proved it separately in Proposition 3.3.7, and made use of it in the proof of the theorem!): if (3.18) is provable from $\mathbb{T}$, then it holds in the universal model U in $\mathcal{C}_{\mathbb{T}}$ by the construction of U,

$$U \models \ \Gamma \mid \varphi \vdash \psi \, .$$

But since regular functors preserve the interpretations of regular formulas $[\![\Gamma \mid \varphi]\!], [\![\Gamma \mid \psi]\!]$ (as well as entailments between them), the entailment (3.18) then holds also in any model M in any regular $\mathcal{C}$, since there is a classifying functor $M^\sharp : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ taking U to M, for which

$$M^\sharp([\![\Gamma \mid \varphi]\!]_U) \cong [\![\Gamma \mid \varphi]\!]_M \, .$$

Completeness follows from the syntactic construction of the universal model U in $\mathcal{C}_{\mathbb{T}}$. The model U is logically generic, in the sense that

$$U \models \ (\Gamma \mid \varphi \vdash \psi) \quad \text{iff} \quad \mathbb{T} \text{ proves } (\Gamma \mid \varphi \vdash \psi) \, .$$

Thus if $\Gamma \mid \varphi \vdash \psi$ holds in all models, then it holds in particular in U, and is therefore provable. □

3.2.5 Coherent logic

A regular category is coherent if all the subobject posets are distributive lattices, and that structure is stable under pullback. We add rules to regular logic to describe this further structure, show that the rules are sound in coherent categories, and extend the results on functorial semantics of the previous section to the coherent case, including the completeness theorem.

Definition 3.2.35. A cartesian category $\mathcal{C}$ is *coherent* if:

1. $\mathcal{C}$ is regular, i.e. it has coequalizers of kernel pairs, and regular epimorphisms are stable under pullback,
2. each subobject poset $\mathsf{Sub}(A)$ has all finite joins, in particular 0 and $U \vee V$,
3. for each map $f : A \to B$, the pullback functor $f^* : \mathsf{Sub}(B) \longrightarrow \mathsf{Sub}(A)$ preserves the joins:
$$f^* 0_B = 0_A, \qquad f^*(U \vee V) = f^*U \vee f^*V\,.$$

Note that since joins are stable under pullback in a coherent category, the meets distribute over the joins,

$$U \wedge (V \vee W) = (U \wedge V) \vee (U \wedge W)\,, \tag{3.19}$$

so that the posets $\mathsf{Sub}(A)$ are distributive lattices. Indeed, this follows from the fact that $U \wedge V$ may be written as

$$U \wedge V = \Sigma_U \circ U^*(V) \tag{3.20}$$

where $\Sigma_U : \mathsf{Sub}(U) \to \mathsf{Sub}(A)$ is the left adjoint (composition) of the pullback functor $U^* : \mathsf{Sub}(A) \to \mathsf{Sub}(U)$ along the inclusion $U \rightarrowtail A$. Since left adjoints preserve colimits, and thus joins, we therefore have

$$U \wedge (V \vee W) = \Sigma_U \circ U^*(V \vee W) = \Sigma_U \circ U^*(V) \vee \Sigma_U \circ U^*(W) = (U \wedge V) \vee (U \wedge W)\,.$$

A category is said to have have *stable sums* if it has all finite coproducts, in particular an initial object 0 and binary coproducts $A+B$, and these are stable under pullback, in the expected sense. The following simple observation provides plenty of examples of coherent categories.

Proposition 3.2.36. *Regular categories with stable sums are coherent.*

Proof. Given subobjects $U, V \rightarrowtail A$, let $U \vee V$ be the image of the canonical map $U+V \to A$ as indicated below.

$$\begin{tikzcd}
 & U+V \arrow[d, two heads] & \\
U \arrow[ur] \arrow[dr, tail] & U \vee V \arrow[d, tail] & V \arrow[ul] \arrow[dl, tail] \\
 & A &
\end{tikzcd}$$

This is easily seem to be the supremum of U and V in $\mathsf{Sub}(A)$. Since the unique map $0 \to A$ is always monic, it determines the subobject $0 \rightarrowtail A$. Thus $\mathsf{Sub}(A)$ has all finite joins, and they are stable by stability of the coproducts and image factorizations. □

As examples of coherent categories we thus have Set and $\mathsf{Set}_{\text{fin}}$, as well as all functor categories $\mathsf{Set}^{\mathbb{C}}$ since limits and colimits (and thus image factorizations) there are computed pointwise.

Exercise 3.2.37. Is the category of H-presets for a heyting algebra H from Section 3.2.1 coherent?

Coherent logic is the extension of regular logic by adding rules corresponding to joins.

Definition 3.2.38. A *coherent theory* $\mathbb{T}$ is (a type theory together with) a set of axioms expressed in the fragment of logic built from $=$, $\top$, $\bot$, $\wedge$, $\vee$, and $\exists$.

We thus extend the formation rules for formulas in context by two additional clauses:

7. The 0-ary relation symbol $\bot$ (pronounced "false") is a formula :

$$\frac{\cdot}{\Gamma \mid \bot \ \mathtt{pred}}$$

8. Disjunction:

$$\frac{\Gamma \mid \varphi \ \mathtt{pred} \qquad \Gamma \mid \psi \ \mathtt{pred}}{\Gamma \mid \varphi \vee \psi \ \mathtt{pred}}$$

(We also again add the evident additional clauses for substitution of terms into formulas.)

A *coherent theory* then consists of axioms of the form

$$\Gamma \mid \varphi \vdash \psi$$

where φ, ψ are *coherent formulas*. Coherent logic not only allows for disjunctions $\varphi \vee \psi$ on both side of the $\vdash$, but the presence of the symbol $\bot$ allows for a certain amount of negation, in the form $\varphi \vdash \bot$, as the following classical example illustrates.

Example 3.2.39. 1. A ring A (with unit 1) is called *local* if it has a unique maximal ideal. This can be captured with two coherent axioms of the form $0 = 1 \vdash \bot$ (to ensure that $0 \neq 1$), and

$$x : A, y : A \mid \exists z : A.\, z(x + y) = 1 \vdash (\exists z : A.\, zx = 1) \vee (\exists z : A.\, zy = 1)$$

2. Another example is the theory of *fields*, which can be axiomatized by again adding to the theory of rings the law $0 = 1 \vdash \bot$, together with the following:

$$x : A \mid \top \vdash x = 0 \vee (\exists y : A.\, xy = 1)$$

which is a clever way of saying that every non-zero element has a multiplicative inverse.

3. An order example is the notion of a *linear order*, which adds to the cartesian theory of posets the *totality* axiom:

$$x : P, y : P \mid x \leq y \vee y \leq q\,.$$

4. For another example of how we can make use of the constant *false* $\bot$ to get the effect of negation, at least for entire axioms, even though the coherent fragment does not include negation, consider the theory of graphs, with two basic sorts E for edges and V for verticies, and two operations $s, t : (E; V)$ for source and target. A graph $G = (E_G, V_G, s_G, t_G)$ is *acyclic* if it satisfies all the finitely many axioms

$$\exists e_1 \dots e_n : E.\, \big(t(e_1) = s(e_2) \wedge \dots \wedge t(e_n) = s(e_1)\big) \vdash \bot\,.$$

The *rules of inference of coherent logic* are those of regular logic (Section 3.2.3), with additional rules for falshood the disjunctions:

10. Falsehood:

$$\frac{}{\bot \vdash \psi}$$

11. Disjunction:

$$\frac{\varphi \vdash \vartheta \qquad \psi \vdash \vartheta}{\varphi \vee \psi \vdash \vartheta} \qquad \frac{\varphi \vee \psi \vdash \vartheta}{\varphi \vdash \vartheta} \qquad \frac{\varphi \vee \psi \vdash \vartheta}{\psi \vdash \vartheta}$$

12. Distributivity:

$$\varphi \wedge (\psi \vee \vartheta) \vdash\ (\varphi \wedge \psi) \vee (\varphi \wedge \vartheta)$$

The latter of course coresponds to the distributive law (3.19); note that the converse can be derived. Like the Frobenius rule, this will be derivable in the extended system of Heyting logic (see Proposition 3.3.14), and could also be made derivable in a suitably formulated system of coherent logic using multi-sequents $\Gamma \mid \varphi_1, \dots, \varphi_n \vdash \psi$.

The *semantics for coherent logic* extends that for regular logic in the expected way: the disjunctive formulas are interpreted as the corresponding joins in the subobject lattices,

$$[\![\Gamma \mid \bot]\!] = 0\,, \qquad\qquad [\![\Gamma \mid \varphi \vee \psi]\!] = [\![\Gamma \mid \varphi]\!] \vee [\![\Gamma \mid \psi]\!]\,.$$

The additional clauses in the proof of soundness are routine. We can then extend the syntactic construction of the regular classifying category $\mathcal{C}_{\mathbb{T}}$ to include all coherent formulas and prove the following extended functorial semantics theorem for models in coherent categories and *coherent functors*, which are defined to be regular functors that preserve all finite joins of subobjects.

Theorem 3.2.40 (Functorial semantics for coherent logic)**.** *For any coherent theory $\mathbb{T}$, the syntactic category $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models in coherent categories. Specifically, for any coherent category $\mathcal{C}$, there is an equivalence of categories, natural in $\mathcal{C}$,*

$$\mathsf{Coh}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C})\,, \tag{3.21}$$

where $\mathsf{Coh}(\mathcal{C}_{\mathbb{T}}, \mathcal{C})$ is the category of coherent functors and natural transformations. In particular, there is a universal model U in $\mathcal{C}_{\mathbb{T}}$.

The corresponding completeness theorem 3.2.34 then holds as well. We leave the routine details to the reader.

Exercise 3.2.41. Extend the functorial semantics theorem 3.2.32 from regular to coherent logic. Specifically, one must determine the components (3.15) of a natural transformation for the extended language of coherent logic.

3.2.6 Freyd embedding theorem

For a coherent theory $\mathbb{T}$, the syntactic construction of the classifying category $\mathcal{C}_{\mathbb{T}}$ means that it is logically generic in the sense that a sequent $(\Gamma \mid \varphi \vdash \psi)$ is $\mathbb{T}$-provable just in case it holds in the universal model U in $\mathcal{C}_{\mathbb{T}}$. The analogue of Corollary 3.2.34 then states the completeness of coherent logic with respect to models in coherent categories. But a stronger statement can also be shown, namely one that restricts the models required to infer provability. Indeed, for (regular and) coherent theories, it suffices to have validity with respect to just the single "standard" category Set, in order to infer provability *for all theories* $\mathbb{T}$. This is a consequence of the following embedding theorem, which can be seen as a categorical version of the Henkin completeness theorem for first-order logic. It plays roughly the same role as did Birkhoff's prime ideal theorem, Lemma 2.9.13, for distributive lattices. And, as in that case, it will be used below to prove a stronger embedding theorem for Heyting categories.

Theorem 3.2.42 (Freyd). *Let $\mathcal{C}$ be a small coherent category. Given any subobject $S \rightarrowtail X$, if $FS \cong FX$ for every coherent functor $F : \mathcal{C} \to \mathsf{Set}$, then $S \cong X$. It follows that every small coherent category $\mathcal{C}$ has a conservative, coherent embedding into a power of set, $\mathcal{C} \rightarrowtail \mathsf{Set}^X$, where for X one can take a (sufficient) set of "models", i.e., coherent functors $\mathcal{C} \to \mathsf{Set}$.*

Proof. To be added later; for now, see [Joh03, D1.5]. □

The result can also be shown for regular categories, and the proof is somewhat easier for that case.

Corollary 3.2.43. *Coherent logic is sound and complete with respect to classical Set-valued semantics. Specifically, for every coherent theory $\mathbb{T}$ and every sequent $\Gamma \mid \varphi \vdash \psi$,*

$$\mathbb{T} \text{ proves } \Gamma \mid \varphi \vdash \psi \qquad \text{iff} \qquad M \models \Gamma \mid \varphi \vdash \psi \text{ for every model } M \text{ in } \mathsf{Set}\,.$$

3.3 Heyting and Boolean categories

In this section we consider coherent categories that also model the universal quantifier $\forall$, in the sense of Section 3.1.4 ; such categories will be seen to model full first-order logic. One could also consider *cartesian* categories modeling $\forall$, without being coherent, and thus modeling the fragment of logic consisting of $u = v, \top, \wedge, \Rightarrow, \forall$, but we will not do so separately.

Definition 3.3.1. A *Heyting category* is a coherent category with universal quantifiers in the sense of Section 3.1.4. Thus for every map $f : A \to B$, the pullback functor $f^* : \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ has a right adjoint,

$$\forall_f : \mathsf{Sub}(A) \to \mathsf{Sub}(B)\,,$$

in addition to the left adjoint $\exists_f : \mathsf{Sub}(A) \to \mathsf{Sub}(B)$ given by taking images.

Note that in a Heyting category, one therefore has both adjoints to pullback along any map $f : A \to B$,

$$\begin{tikzcd} \mathsf{Sub}(A) \arrow[r, shift left=2, "\exists_f"] \arrow[r, shift right=2, "\forall_f"'] & \mathsf{Sub}(B) \arrow[l, "f^*" description] \end{tikzcd} \qquad \exists_f \dashv f^* \dashv \forall_f\,. \tag{3.22}$$

Moreover, the Beck-Chevalley conditions from Section 3.1.4 are satisfied for both $\exists_f$ (by Proposition 3.2.15) and $\forall_f$ (by Proposition 3.1.28).

A common way to get a Heyting structure on a category $\mathcal{C}$ is when the operation of universal quantification on the subobject lattices $\mathsf{Sub}(A)$ is inherited from a related one on the slice categories $\mathcal{C}/A$; this happens e.g. when $\mathcal{C}$ is *locally cartesian closed*. Recall that a cartesian closed category is a category that has products and exponentials. A category is locally cartesian closed when every slice is cartesian closed.

Definition 3.3.2. A category $\mathcal{C}$ is *locally cartesian closed (lccc)* when it has a terminal object and every slice $\mathcal{C}/A$ is cartesian closed.

Note that every slice category $\mathcal{C}/A$ has a terminal object, namely the identity morphism $1_A : A \to A$, and all $\mathcal{C}/A$ have binary products if, and only if, $\mathcal{C}$ has pullbacks. Thus a locally cartesian closed category has all finite limits because it has a terminal object and pullbacks. In addition, a locally cartesian closed category is cartesian closed because $\mathcal{C} \cong \mathcal{C}/1$.

We describe how exponentials in a slice $\mathcal{C}/A$ can be computed in terms of *change of base functors* and *dependent products*. Given a morphism $f : A \to B$ in $\mathcal{C}$, the "change of base along f" is the pullback functor

$$f^* : \mathcal{C}/B \to \mathcal{C}/A\,.$$

A right adjoint to f^*, when it exists, is called a *dependent product along* f, denoted

$$\Pi_f : \mathcal{C}/A \to \mathcal{C}/B\,.$$

Now an exponential of $b : B \to A$ and $c : C \to A$ in $\mathcal{C}/A$ can be computed in terms of Π_b and b^*. For any $d : D \to A$, we have $b \times_A d = (b^*d) \circ b = \Sigma_b(b^*d)$, hence

$$\begin{array}{c} b \times_A d \to c \\ \hline\hline \Sigma_b(b^*d) \to c \\ \hline\hline b^*d \to b^*c \\ \hline\hline d \to \Pi_b(b^*c) \end{array}$$

Therefore, $c^b = \Pi_b(b^* c)$.

We have proved that if a cartesian category $\mathcal{C}$ has dependent product $\Pi_f : \mathcal{C}/A \to \mathcal{C}/B$ along every morphism $f : A \to B$ then it is locally cartesian closed. The converse holds as well, that is every lccc has dependent products. For a proof see Section **??** or [Awo10, 9.20].

Proposition 3.3.3. *A category $\mathcal{C}$ with a terminal object is locally cartesian closed if, and only if, for any $f : A \to B$ the change of base functor $f^* : \mathcal{C}/B \to \mathcal{C}/A$ has a right adjoint $\Pi_f : \mathcal{C}/A \to \mathcal{C}/B$.*

Proposition 3.3.4. *In an lccc $\mathcal{C}$, for any $f : A \to B$ the change of base functor $f^* : \mathcal{C}/B \to \mathcal{C}/A$ preserves the ccc structure.*

Proof. We need to show that f^* preserves terminal objects, binary products, and exponentials in slices. Because f^* is a right adjoint it preserves limits, hence it preserves terminal objects and binary products. To see that it preserves exponentials we first show that $f^* \circ \Pi_g \cong \Pi_{f^*g} \circ (g^* f)^*$ for $g : C \to B$. Given any $d : D \to C$, and $e : E \to A$:

$$\begin{array}{c}
e \to f^*(\Pi_g d) \\
\hline\hline
\Sigma_f e \to \Pi_g d \\
\hline\hline
g^*(\Sigma_f e) \to d \\
\hline\hline
g^*(f \circ e) \to d \\
\hline\hline
(g^* f) \circ ((f^* g)^* e) \to d \\
\hline\hline
(f^* g)^* e \to (g^* f)^* d \\
\hline\hline
e \to \Pi_{f^* g}((g^* f)^* d)
\end{array}$$

By the Yoneda Lemma it follows that $f^*(\Pi_g d) \cong \Pi_{f^* g}((g^* f)^* d)$. Now we have, for any $c : C \to B$ and $d : D \to B$,

$$f^* c^d = f^*(\Pi_d(d^* c)) = \Pi_{f^* d}((d^* f)^*(d^* c)) = \Pi_{f^* d}((f^* d)^*(f^* c)) = (f^* c)^{(f^* d)} \,.$$

□

Exercise 3.3.5. In the preceding proof we used the fact that $(d^* f)^*(d^* c) \cong (f^* d)^*(f^* c)$ and $g^*(f \circ e) \cong (g^* f) \circ ((f^* g)^* e)$. Prove that this is really so.

Locally cartesian closed categories are an important example of categories with universal quantifiers.

Proposition 3.3.6. *A locally cartesian closed category has universal quantifiers.*

Proof. Suppose $\mathcal{C}$ is locally cartesian closed. First observe that a morphism $m : M \to A$ is mono if, and only if, the morphism

$$\begin{tikzcd}
M \arrow[rr, "m"] \arrow[dr, "m"'] & & A \arrow[dl, "1_A"] \\
& A &
\end{tikzcd}$$

is mono in $\mathcal{C}/A$. Because right adjoints preserve monos, $\Pi_f : \mathcal{C}/A \to \mathcal{C}/B$ preserve monos for any $f : A \to B$, that is, if $m : M \rightarrowtail A$ is mono then $\Pi_f m : \Pi_f M \to B$ is mono in $\mathcal{C}$. Therefore, we may define $\forall_f$ as the restriction of Π_f to $\mathsf{Sub}(A)$. To be more precise, a subobject $[m : M \rightarrowtail A]$ is mapped by $\forall_f$ to the subobject $[\Pi_f m : \Pi_f M \rightarrowtail B]$. This works because for any monos $m : M \rightarrowtail A$ and $n : N \rightarrowtail B$ we have

$$\begin{array}{rr} f^*[m : M \rightarrowtail A] \le [n : N \rightarrowtail B] & \text{in } \mathsf{Sub}(B) \\ \hline\hline f^* m \to n & \text{in } \mathcal{C}/B \\ \hline\hline m \to \Pi_f n & \text{in } \mathcal{C}/A \\ \hline\hline [m] \le \forall_f [n] & \text{in } \mathsf{Sub}(A) \end{array}$$

The Beck-Chevalley condition for $\forall_f$ follows from Proposition 3.3.4. Indeed, if $g : C \to B$ and $m : M \rightarrowtail C$ then

$$f^*(\Pi_g m) \cong \Pi_{f^* g}((g^* f)^* m) \;,$$

therefore

$$f^*(\forall_g [m : M \rightarrowtail C]) = \forall_{f^* g}((g^* f)^* [m : M \rightarrowtail C]) \;,$$

as required. □

Summarizing, diagram (3.23), which may be called *Lawvere's hyperdoctrine diagram*, displays the relation between the quantifiers and the change of base functors.

$$\begin{tikzcd}[row sep=huge, column sep=huge]
\mathcal{C}/A \arrow[r, shift left=2, "\Sigma_f"] \arrow[r, shift right=2, "\Pi_f"'] \arrow[d, shift right, "\mathsf{im}"'] & \mathcal{C}/B \arrow[l, "f^*" description] \arrow[d, shift right, "\mathsf{im}"'] \\
\mathsf{Sub}(A) \arrow[r, shift left=2, "\exists_f"] \arrow[r, shift right=2, "\forall_f"'] \arrow[u, hook, shift right, "I"'] & \mathsf{Sub}(B) \arrow[l, "f^*" description] \arrow[u, hook, shift right, "I"']
\end{tikzcd} \tag{3.23}$$

In Section 3.3.3 below we shall see that all presheaf categories $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ are Heyting, and therefore have universal quantifiers, which we will compute explicitly (they are *not* pointwise!).

3.3.1 Heyting logic

We can now extend the *formation rules* for the logical language to include universally quantified formulas in the expected way:

$$\frac{\Gamma, x : A \mid \varphi \ \mathtt{pred}}{\Gamma \mid \forall x : A.\, \varphi \ \mathtt{pred}}$$

The corresponding additional *rule of inference* for the universal quantifier is:

$$\frac{y : B, x : A \mid \vartheta \vdash \varphi}{\overline{y : B \mid \vartheta \vdash \forall x : A.\, \varphi}}$$

Note that the lower judgement is well-formed only if $x : A$ does not occur freely in ϑ.

Finally, we extend the *interpretation* from coherent formulas from (Section 3.2.5) to formulas including universal quantifiers by the additional clause for $\forall x : A.\, \varphi$ using the universal quantifiers in the Heyting category,

$$[\![\Gamma \mid \forall x : A.\, \varphi]\!] = \forall_A [\![\Gamma, x : A \bigm| \varphi]\!] \;,$$

where

$$\forall_A = \forall_\pi : \mathsf{Sub}([\![\Gamma]\!] \times [\![A]\!]) \to \mathsf{Sub}([\![\Gamma]\!])$$

is the universal quantifier along the projection $\pi : [\![\Gamma]\!] \times [\![A]\!] \to [\![\Gamma]\!]$.

The following is then immediate from the results of section **??**.

Proposition 3.3.7. *The rules for the universal quantifier are sound with respect to the interpretation in Heyting categories.*

Implication

Recall that the rules of inference for implication state that $\Rightarrow$ is right adjoint to $\wedge$:

$$\frac{\Gamma \mid \vartheta \ \texttt{pred} \qquad \Gamma \mid \varphi \ \texttt{pred}}{\Gamma \mid (\vartheta \Rightarrow \varphi) \ \texttt{pred}} \qquad\qquad \frac{\Gamma \mid \psi \wedge \vartheta \vdash \varphi}{\overline{\Gamma \mid \psi \vdash \vartheta \Rightarrow \varphi}}$$

Exercise 3.3.8. Show that the above two-way rule can be replaced by the following introduction and elimination rules:

$$\frac{\Gamma \mid \psi \wedge \vartheta \vdash \varphi}{\Gamma \mid \psi \vdash \vartheta \Rightarrow \varphi} \qquad\qquad \frac{\Gamma \mid \psi \vdash \vartheta \Rightarrow \varphi \qquad \Gamma \mid \psi \vdash \vartheta}{\Gamma \mid \psi \vdash \varphi}$$

If we want to interpret implication in a Heyting category $\mathcal{C}$ we therefore require $\mathsf{Sub}(A)$ to be Cartesian closed for every $A \in \mathcal{C}$. However, we must not forget that implication interacts with substitution by the rule

$$(\vartheta \Rightarrow \varphi)[t/x] \;=\; \vartheta[t/x] \Rightarrow \varphi[t/x] \;.$$

Semantically this means that implication is *stable* under pullbacks.

Definition 3.3.9. A cartesian category $\mathcal{C}$ has *implications* when, for every $A \in \mathcal{C}$, the poset $\mathsf{Sub}(A)$ is cartesian closed, with stable implication $\Rightarrow$. This means that for $U, V \in \mathsf{Sub}(A)$ and $f : B \to A$,

$$f^*(U \Rightarrow V) = (f^*U \Rightarrow f^*V) \;.$$

Proposition 3.3.10. *If a cartesian category has universal quantifiers then it has implications.*

Proof. Let $[u : U \rightarrowtail A]$ and $[v : V \rightarrowtail A]$ be subobjects of A. Define

$$([u] \Rightarrow [v]) = \forall_u(u^*[v]) \ ,$$

as indicated below

$$\begin{tikzcd}
u^*([v]) \arrow[r, tail] \arrow[d, tail] & V \arrow[d, tail, "v"] & \forall_u u^*([v]) \arrow[dl, tail] \\
U \arrow[r, tail, "u"'] & A &
\end{tikzcd}$$

Then for any subobject $[w : W \rightarrowtail A]$ we have:

$$\begin{array}{ll}
[w] \leq [u] \Rightarrow [v] & \text{in } \mathsf{Sub}(A) \\
\hline\hline
[w] \leq \forall_u(u^*[v]) & \text{in } \mathsf{Sub}(A) \\
\hline\hline
u^*[w] \leq u^*[v] & \text{in } \mathsf{Sub}(U) \\
\hline\hline
\exists_u(u^*w) \leq v & \text{in } \mathsf{Sub}(A) \\
\hline\hline
[u] \wedge [w] \leq [v] & \text{in } \mathsf{Sub}(A)
\end{array}$$

Note that we used the decomposition of $[u] \wedge [w]$ as $\exists_u(u^*w)$ from (3.20).

Finally, stability of $\Rightarrow$ follows from Beck-Chevalley condition for $\forall$. □

Exercise 3.3.11. Prove the last claim of the proof.

Corollary 3.3.12. *Any LCCC has universal quantifiers and implications.*

Negation

In any Heyting category, we have not only implications $U \Rightarrow V$ making each $\mathsf{Sub}(A)$ cartesian closed, but also 0 and $\vee$ coming from the coherent structure, so that $\mathsf{Sub}(A)$ is a Heyting algebra. Here 0 is the bottom element $[0 \rightarrowtail A]$, and $\vee$ is the join $[p \vee q \rightarrowtail A]$, in the poset $\mathsf{Sub}(A)$. We can therefore also define *negation* $\neg U$ as usual in a Heyting algebra, namely:

$$\neg U \ = \ (U \Rightarrow 0)\,, \tag{3.24}$$

These negations are stable under pullback because the Heyting implications and the bottom element 0 are stable.

We can therefore add *formulas* with negation to the logical language, along with the evident two-way *rule of inference*:

$$\frac{\Gamma \mid \varphi \ \texttt{pred}}{\Gamma \mid \neg\varphi \ \texttt{pred}} \qquad\qquad \begin{array}{c} \Gamma \mid \vartheta \vdash \neg\varphi \\ \hline\hline \Gamma \mid \vartheta \wedge \varphi \vdash \bot \end{array}$$

We give negated formulas the obvious *interpretation*: given $[\![\varphi]\!]$ in $\mathsf{Sub}(A)$, we set

$$[\![\neg\varphi]\!] \;=\; \neg[\![\varphi]\!] \;=\; [\![\varphi]\!] \Rightarrow 0\,.$$

using the Heyting implication $\Rightarrow$ and bottom element 0 in $\mathsf{Sub}(A)$. The following is then immediate.

Proposition 3.3.13. *The rules for negation are sound in any Heyting category.*

Given Heyting implication, we can prove the distributivity rule from Section 3.2.5 for conjunction and disjunction.

Proposition 3.3.14. *The distributivity rule is provable in Heyting logic:*

$$\varphi \wedge (\psi \vee \vartheta) \;\vdash\; (\varphi \wedge \psi) \vee (\varphi \wedge \vartheta)$$

Proof.

$$\begin{array}{c}
(\varphi \wedge \psi) \vee (\varphi \wedge \vartheta) \vdash \zeta \\ \hline\hline
(\varphi \wedge \psi) \vdash \zeta \qquad (\varphi \wedge \vartheta) \vdash \zeta \\ \hline\hline
\psi \vdash \varphi \Rightarrow \zeta \qquad \vartheta \vdash \varphi \Rightarrow \zeta \\ \hline\hline
\psi \vee \vartheta \vdash \varphi \Rightarrow \zeta \\ \hline\hline
\varphi \wedge (\psi \vee \vartheta) \vdash \zeta
\end{array}$$

Thus, in fact,

$$\varphi \wedge (\psi \vee \vartheta) \;\dashv\vdash\; (\varphi \wedge \psi) \vee (\varphi \wedge \vartheta).$$

□

Perhaps more surprisingly, given universal quantifiers, we can actually prove the Frobenius rule from Section 3.2.3 for existential quantifiers.

Proposition 3.3.15. *The Frobenius rule is provable in Heyting logic:*

$$(\exists y : B.\, \varphi) \wedge \psi \;\vdash\; \exists y : B.\, (\varphi \wedge \psi)$$

provided the variable $y : B$ does not occur freely in ψ.

Proof.

$$\begin{array}{c}
\exists y : B.\, (\varphi \wedge \psi) \vdash \zeta \\ \hline\hline
y : B \mid \varphi \wedge \psi \vdash \zeta \\ \hline\hline
y : B \mid \varphi \vdash \psi \Rightarrow \zeta \\ \hline\hline
(\exists y : B.\, \varphi) \vdash \psi \Rightarrow \zeta \\ \hline\hline
(\exists y : B.\, \varphi) \wedge \psi \;\vdash \zeta
\end{array}$$

Thus, in fact,

$$(\exists y : B.\varphi) \wedge \psi \;\dashv\vdash\; \exists y : B.(\varphi \wedge \psi).$$

□

$$\frac{}{\bot \vdash \varphi} \qquad\qquad \frac{}{\varphi \vdash \top}$$

$$\begin{array}{c} \varphi \vdash \vartheta \qquad \psi \vdash \vartheta \\ \hline\hline \varphi \vee \psi \vdash \vartheta \end{array} \qquad\qquad \begin{array}{c} \vartheta \vdash \varphi \qquad \vartheta \vdash \psi \\ \hline\hline \vartheta \vdash \varphi \wedge \psi \end{array}$$

$$\begin{array}{c} \vartheta \wedge \varphi \vdash \psi \\ \hline\hline \vartheta \vdash \varphi \Rightarrow \psi \end{array}$$

$$\begin{array}{c} x : A \mid \varphi \vdash \vartheta \\ \hline\hline \exists\, x : A \,.\, \varphi \vdash \vartheta \end{array} \qquad\qquad \begin{array}{c} x : A \mid \vartheta \vdash \varphi \\ \hline\hline \vartheta \vdash \forall\, x : A \,.\, \varphi \end{array}$$

Figure 3.2: Adjoint rules of inference for Heyting logic

Exercise 3.3.16. In classical logic, one has the *de Morgen laws* for negation,

$$\neg(\varphi \wedge \psi) \dashv\vdash \neg\varphi \vee \neg\psi$$
$$\neg(\varphi \vee \psi) \dashv\vdash \neg\varphi \wedge \neg\psi$$

Which of these four entailments can you prove in Heyting logic?

Adjoint rules of Heyting logic

Figure 3.2 collects the rules of inference for Heyting logic. These are stated as two-way rules to emphasize the respective underlying adjunctions. The rules for disjunction and conjunction in the bottom-up direction are, of course, to be understood a two separate rules, left and right. The contexts are omitted where there is no change between the top and bottom, thus e.g. the rule for existential quantifier can be stated in full as:

$$\begin{array}{c} \Gamma, x : A \mid \varphi \vdash \vartheta \\ \hline\hline \Gamma \mid \exists\, x : A \,.\, \varphi \vdash \vartheta \end{array}$$

Negation $\neg\,\varphi$ is treated as a defined by

$$\neg\,\varphi \;:=\; \varphi \Rightarrow \bot\,.$$

It therefore satisfies the derived rule:

$$\begin{array}{c} \vartheta \wedge \varphi \vdash \bot \\ \hline\hline \vartheta \vdash \neg\,\varphi \end{array}$$

The rules for *equality*, recall from Section 3.1.3, were:

$$\frac{}{\psi \vdash t =_A t} \qquad\qquad \frac{\psi \vdash t =_A u \qquad \psi \vdash \varphi[t/z]}{\psi \vdash \varphi[u/z]} \tag{3.25}$$

Lawvere [Law70] observed that equality can also be seen as an adjoint, namely to the operation of pullback along the diagonal $\Delta : A \to A \times A$ in any cartesian category. Indeed, we have an adjunction

$$\begin{tikzcd}
\mathsf{Sub}(A) \arrow[d, shift right, "\exists_\Delta"'] \\
\mathsf{Sub}(A \times A) \arrow[u, shift right, "\Delta^*"']
\end{tikzcd}
\qquad
\begin{array}{c}
x : A \mid \vartheta(x) \vdash \varphi(x, x) \\
\hline\hline
x : A, y : A \mid (x = y) \wedge \vartheta(x) \vdash \varphi(x, y)
\end{array}
\tag{3.26}$$

where we have displayed the variables in the style $\varphi(x, y)$ in order to emphasize the effect of Δ^* as a "contraction of variables",

$$\Delta^*(\varphi(x, y)) = \varphi(x, x)\,.$$

The effect of the left adjoint $\exists_\Delta$ (which is simply composition with Δ, because it is monic) is given by

$$\exists_\Delta(\vartheta(x)) \ = \ \big(x = y \wedge \vartheta(x)\big)\,.$$

The adjoint rule (3.26) may be called *Lawvere's Law.* It is equivalent to the standard rules (3.25).

Exercise 3.3.17. Prove the equivalence of (3.25) and (3.26).

We state the following for the record as a summary of the foregoing discussion.

Proposition 3.3.18 (Soundness)**.** *The adjoint rules of inference for Heyting logic as stated in Figure 3.2, as well as Lawvere's Law* (3.26)*, are sound in any Heyting category.*

Theorem 3.3.22 implies that these rules are also complete with respect to models in Heyting categories.

3.3.2 First-order logic

Heyting logic with equality is often called *intuitionistic first-order logic* (IFOL). It lacks the classical laws of excluded middle $\varphi \vee \neg\varphi$ and double negation elimination $\neg\neg\varphi \Rightarrow \varphi$, but adding either one of these implies the other (proof!), and gives a system equivalent to standard first-order logic – with one exception: one still cannot prove the classical law

$$\forall x : A .\, \varphi \vdash \exists x : A \,.\varphi\,. \tag{3.27}$$

The latter law, which is satisfied only in non-empty domains, is considered by some to be a defect of the conventional formulation of first-order logic. It would follow if we were to forget about the contexts, essentially permitting inferences of the form

$$\frac{x : A \mid \varphi \vdash \psi}{\cdot \mid \varphi \vdash \psi} \tag{3.28}$$

when $x : A$ does not occur freely in φ or ψ (cf. Remark 3.1.29).

Exercise 3.3.19. Assume the rule (3.28) and prove the entailment (3.27).

Any conventional first-order theory can be formulated in IFOL, often in more than one way, since classical logic may collapse differences between concepts that are intuitionistically distinct (like, most simply, φ and $\neg\neg\varphi$). Our interest in intuitionistic logic does not arise from any philosophical scruples about the validity of the classical laws of excluded middle or double negation, but rather the fact that the logic of variable structures is naturally intuitionistic, as we will see in Section **??**.

Example 3.3.20. An example of a first-order theory that is not (immediately) coherent is the theory of dense linear orders. In addition to the poset axioms, and the totality axiom $x, y : P \mid \top \vdash (x \leq y \vee y \leq x)$, one adds density e.g. in the form

$$x, y : P \mid (x \leq y \wedge x \neq y) \vdash (\exists z : P.\, x \leq z \wedge x \neq z \wedge z \leq y \wedge z \neq y)\,.$$

The classifying category of an intuitionistic first-order theory

Given a theory $\mathbb{T}$ in IFOL, we can build the syntactic category $\mathcal{C}_{\mathbb{T}}$ from the formulas over $\mathbb{T}$, as was done for coherent logic in Section 3.2.4. The objects again have the form $[\Gamma \mid \varphi]$, but now using the Heyting formulas φ, including the logical operations $\forall$, and $\Rightarrow$. The result will then be a coherent category with universal quantifiers, and thus a Heyting category in the sense of Definition 3.3.1. Given another Heyting category $\mathcal{C}$ with a $\mathbb{T}$-model $M \in \mathsf{Mod}(\mathbb{T}, \mathcal{C})$, the interpretation $[\![-]\!]_M$ associated to the model M determines a Heyting functor,

$$M^\sharp : \mathcal{C}_{\mathbb{T}} \longrightarrow \mathcal{C} \tag{3.29}$$

$$[\Gamma \mid \varphi] \longmapsto [\![\Gamma \mid \varphi]\!]_M \tag{3.30}$$

We would like to show that $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models, in the sense that this assignment determines an equivalence of categories, associating homomorphisms of $\mathbb{T}$-models $h : M \to N$ in the category $\mathsf{Mod}(\mathbb{T}, \mathcal{C})$, and natural transformations of the associated classifying Heyting functors $M^\sharp \to N^\sharp$ in $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$.

However, there is a problem. Reviewing the proof of Theorem 3.2.32, we needed to show that definable subobjects are natural in model homomorphisms, in the following sense: let $F, G : \mathcal{C}_{\mathbb{T}} \longrightarrow \mathcal{C}$ be functors classifying models FU and GU, and let $h : FU \to GU$ be a model homomorphism. We have maps $h_A : F(A) \longrightarrow G(A)$ for all basic types $A = [x : A \mid \top]$, commuting with the interpretations of the function symbols f and the basic relations R. For each object $[x : A \mid \varphi]$, say, the components

$$h_{[x:A|\varphi]} : F[x : A \mid \varphi] = [\![x : A \mid \varphi]\!]_{FU} \longrightarrow [\![x : A \mid \varphi]\!]_{GU} = G[x : A \mid \varphi]$$

were then defined on definable subobject $[\![x : A \mid \varphi]\!]_{FU} \rightarrowtail [\![A]\!]_{FU} = FA$, in such a way that the following diagram commutes as indicated.

$$\begin{tikzcd}
\llbracket x : A \mid \varphi \rrbracket_{FU} \arrow[r, tail] \arrow[d, dotted, "h_{[x:A|\varphi]}"'] & \llbracket A \rrbracket_{FU} \arrow[d, "h_A"] \\
\llbracket x : A \mid \varphi \rrbracket_{GU} \arrow[r, tail] & \llbracket A \rrbracket_{GU}
\end{tikzcd} \tag{3.31}$$

This we could do for all *coherent* formulas φ, as was shown by induction on the structure of φ. However, this is no longer possible when φ is Heyting. Most simply, if $\varphi = \neg\psi$ for coherent ψ, there is no need for the following to commute on the left.

$$\begin{tikzcd}
\llbracket x : A \mid \neg\psi \rrbracket_{FU} \arrow[r, tail] \arrow[d, dotted, "h_{[x:A|\neg\psi]}"'] & \llbracket A \rrbracket_{FU} \arrow[d, "h_A"] & \llbracket x : A \mid \psi \rrbracket_{FU} \arrow[l, tail] \arrow[d, "h_{[x:A|\psi]}"] \\
\llbracket x : A \mid \neg\psi \rrbracket_{GU} \arrow[r, tail] & \llbracket A \rrbracket_{GU} & \llbracket x : A \mid \psi \rrbracket_{GU} \arrow[l, tail]
\end{tikzcd} \tag{3.32}$$

Very concretely, let $\mathbb{T}$ be the theory of groups, FU and GU groups in Set and $h_A : \llbracket A \rrbracket_{FU} \to \llbracket A \rrbracket_{GU}$ the trivial homomorphism that takes everything $a \in \llbracket A \rrbracket_{FU}$ to the unit $e_{GU} \in \llbracket A \rrbracket_{GU}$, and ψ the formula $x : A \mid x = e$. Then $\llbracket x : A \mid \psi \rrbracket_{GU} = \{e_{GU}\}$ and so $\llbracket x : A \mid \neg\psi \rrbracket_{GU} = \{y \in \llbracket A \rrbracket_{GU} \mid y \neq e_{GU}\}$, so there is a factorization $h_{[x:A|\neg\psi]} : \llbracket x : A \mid \neg\psi \rrbracket_{FU} \to \llbracket x : A \mid \neg\psi \rrbracket_{GU}$ only if FU is trivial.

The same holds, of course, for subobjects defined by the other Heyting operations, such as $[x : A \mid \vartheta \Rightarrow \psi]$ and $[x : A \mid \forall y : B.\psi]$; there need not be any factorizations $h_{[x:A|\varphi]}$ as indicted in (3.31).

Our solution (although not the only possible one) is to consider only *isomophisms of models* $h : M \cong N$ and *natural isomorphisms* between the classifying functors.

Lemma 3.3.21. *In the situation of diagram* (3.31)*, if the model homomorphism* $h : FU \to GU$ *is an isomorphism, then for any Heyting formula* $[\Gamma \mid \varphi]$ *there is a unique factorization*

$$h_{[\Gamma|\varphi]} : F[\Gamma \mid \varphi] = \llbracket x : A \mid \varphi \rrbracket_{FU} \longrightarrow \llbracket x : A \mid \varphi \rrbracket_{GU} = G[x : A \mid \varphi]$$

making the corresponding diagram (3.31) *commute.*

Proof. Induction on φ. □

Now for every Heyting category $\mathcal{C}$, let us define $\mathsf{Mod}(\mathbb{T}, \mathcal{C})^i$ to be the category of $\mathbb{T}$-models in $\mathcal{C}$, and their isomorphisms; thus $\mathsf{Mod}(\mathbb{T}, \mathcal{C})^i$ is a groupoid. Accordingly we let $\mathsf{Heyt}(\mathcal{C}_{\mathbb{T}}, \mathcal{C})^i$ to be the category of all Heyting functors $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ and natural *iso*morphisms between them – thus also a groupoid. Then just as in previous cases we can show:

Theorem 3.3.22 (Functorial semantics for intuitionistic first-order logic)**.** *For any theory* $\mathbb{T}$ *in (intuitionistic) first-order logic, the syntactic category* $\mathcal{C}_{\mathbb{T}}$ *classifies* $\mathbb{T}$*-models in*

Heyting categories. Specifically, for any Heyting category $\mathcal{C}$, there is an equivalence of categories, natural in $\mathcal{C}$,

$$\mathsf{Heyt}(\mathcal{C}_{\mathbb{T}}, \mathcal{C})^i \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{C})^i\,, \tag{3.33}$$

where $\mathsf{Heyt}(\mathcal{C}_{\mathbb{T}}, \mathcal{C})^i$ is the groupoid of Heyting functors and natural isomorphisms, and $\mathsf{Mod}(\mathbb{T}, \mathcal{C})^i$ is the groupoid of $\mathbb{T}$-models in $\mathcal{C}$. In particular, there is a universal model U in $\mathcal{C}_{\mathbb{T}}$.

The corresponding completeness theorem 3.2.34 for intuitionistic first-order logic with respect to models in Heyting categories then holds as well. We leave the routine details to the reader.

Boolean categories

A Boolean category may be defined as a coherent category in which every subobject $U \rightarrowtail A$ is *complemented*, in the sense that it there is some (necessarily unique) $V \rightarrowtail A$ such that $U \wedge V \leq 0$ and $1 \leq U \vee V$ in $\mathsf{Sub}(A)$. One can then introduce the Boolean negation $\neg U = V$, and show that each $\mathsf{Sub}(A)$ is a Boolean algebra. Indeed one can then show that every Boolean category is Heyting, using the familiar definitions $\forall\varphi = \neg\exists\neg\varphi$ and $\varphi \Rightarrow \psi = \neg\varphi \vee \psi$.

This definition, however, leads to the wrong notion of a "Boolean classifying category", for the reasons just discussed with respect to Heyting categories: although every coherent functor between Boolean categories is Boollean, the natural transformations between classifying functors will not be simply the homomorphisms. (They will be something interesting, namely elementary embeddings, but we shall not pursue this further here; see [**?**].) Thus it seems preferable for our purposed to define a Boolean category to be a Heyting category with complemented subobjects:

Definition 3.3.23. A Heyting catgeory $\mathcal{C}$ is *Boolean* if every subobject lattice $\mathsf{Sub}(A)$ is a Booean algebra. Thus for all subobjects $U \rightarrowtail A$, the Heyting complement $\neg U$ satisfies $U \vee \neg U = 1$ in $\mathsf{Sub}(A)$.

Of course, the category Set is Boolean. A presheaf category $\mathsf{Set}^{\mathbb{C}}$ is in general *not* Boolean, but an important special case always is, namely when $\mathbb{C}$ is a groupoid. (Set^G is called the *category of G-sets.*)

Exercise 3.3.24. Regard a group G as a category with one object. Show that in the functor category Set^G, every subobject lattice $\mathsf{Sub}(A)$ is a Boolean algebra.

The classifying category theorem 3.3.22 for Heyting categories, and indeed the entire framework of functorial semantics, applies *mutatis mutandis* to classical first-order logic and Boolean categories. We will not spell out the details, which do not differ in any unexpected way from the more general Heyting case.

Exercise 3.3.25. Assume that $\mathcal{C}$ is coherent and has complemented subobjects in the sense just defined. Prove that then each $\mathsf{Sub}(A)$ is a Boolean algebra, and that $\mathcal{C}$ is a Heyting category.

Exercise 3.3.26. Show that a Heyting category $\mathcal{C}$ is Boolean if, and only if, in each $\mathsf{Sub}(A)$ the Heyting complement $\neg U$ always satisfies $\neg\neg U = U$.

3.3.3 Examples

Sets. The category Set is of course complete and cocomplete. It is cartesian closed, with function sets $B^A = \{f : A \to B\}$ as exponentials. It is also locally cartesian closed, because the slice category Set/I is equivalent to the category Set^I of I-indexed families of sets $(A_i)_{i\in I}$, for which the exponentials can be computed pointwise: for $A = (A_i)_{i\in I}$ and $B = (B_i)_{i\in I}$ we can set $B^A = (B_i{}^{A_i})_{i\in I}$. Since pullback is therefore a left adjoint, regular epis are stable and so Set is coherent. It is then Heyting by Proposition 3.3.6.

In order to compute the Heyting structure explicitly, consider any map $f : A \to B$ and the resulting adjunctions from (3.22),

$$\begin{tikzcd} \mathsf{Sub}(A) \arrow[r, bend left, "\exists_f"] \arrow[r, bend right, "\forall_f"'] & \mathsf{Sub}(B) \arrow[l, "f^*" description] \end{tikzcd} \qquad \exists_f \dashv f^* \dashv \forall_f\,.$$

For $U \in \mathsf{Sub}(A)$ and $V \in \mathsf{Sub}(B)$ we then have:

$$\begin{aligned} f^*(V) &= f^{-1}(V) = \{a \in A \mid f(a) \in V\} \\ \exists_f(U) &= \{b \in B \mid \text{for some } a \in f^{-1}\{b\}, a \in U\} \\ \forall_f(U) &= \{b \in B \mid \text{for all } a \in f^{-1}\{b\}, a \in U\} \end{aligned} \tag{3.34}$$

It follows that in Set the implications $U \Rightarrow V$ for $U, V \in \mathsf{Sub}(A)$ have the form

$$\begin{aligned} (U \Rightarrow V) &= \{a \in A \mid a \in U \text{ implies } a \in V\} \\ &= (A \setminus U) \cup V\,. \end{aligned}$$

For negation, we then have

$$\begin{aligned} \neg\, U &= \{a \in A \mid a \notin U\} \\ &= (A \setminus U)\,, \end{aligned}$$

as expected. Of course, Set is Boolean.

Exercise 3.3.27. In Set consider the dependent sum and product along the unique function $I \to 1$. Show that for $a : A \to I$ the set $\Pi_I A$ is the set of right inverses of a:

$$\Pi_I A = \big\{ s : I \to A \;\big|\; a \circ s = 1_I \big\}\ .$$

If $(A_i)_{i\in I}$ is a family of sets indexed by I and we take

$$A = \textstyle\coprod_{i\in I} A_i = \big\{ \langle i, x\rangle \in I \times \bigcup_{i\in I} A_i \;\big|\; i \in I \;\&\; x \in A_i \big\}$$

with $a = \pi_0 : \langle i, x\rangle \mapsto i$ then $\Pi_{!_I} A$ is precisely the cartesian product $\Pi_{i\in I} A_i$. Calculate what Π_f is in Set for a general $f : J \to I$, and conclude that Set is locally cartesian closed.

Presheaves. For a small category $\mathbb{C}$, the presheaf category $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ has pointwise limits and colimits and is cartesian closed with the exponential of presheaves P, Q calculated using Yoneda as,

$$Q^P(C) \cong \mathsf{Hom}(\mathsf{y}C, Q^P) \cong \mathsf{Hom}(\mathsf{y}C \times P, Q)\,, \qquad \text{for } C \in \mathbb{C}.$$

But then $\widehat{\mathbb{C}}$ is also LCC, because for any presheaf P, the slice category $\widehat{\mathbb{C}}/P$ is equivalent to presheaves on the *category of elements* $\int_{\mathbb{C}} P$,

$$\widehat{\mathbb{C}}/P \;=\; (\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}})/P \;\simeq\; \mathsf{Set}^{(\int_{\mathbb{C}} P)^{\mathsf{op}}}\,.$$

See [Awo10, 9.23].

We first consider the poset $\mathsf{Sub}(P)$ for any presheaf P on $\mathbb{C}$. Let $U \rightarrowtail P$ be any subobject, then since monos in are pointwise in $\widehat{\mathbb{C}}$, and they are represented by subsets in Set, we can represent U by a family $UC \subseteq PC$ of subsets. If $f : P \to Q$ is a natural transformation, the inverse image of $V \rightarrowtail Q$ can then be calculated pointwise from $f_C : PC \to QC$ as

$$f^*(V)(C) = f_C^{-1}(VC) = \{x \in PC \mid f_C(x) \in VC\}\,.$$

The image $\exists_f(U)$, as a coequalizer, is also pointwise, therefore

$$\exists_f(U)(C) = \{y \in QC \mid \text{for some } x \in f_C^{-1}\{y\},\, x \in UC\}\,.$$

The direct image $\forall_f(U)$ is however *not pointwise*, so we must determine it directly. The problem with the obvious attempt

$$\forall_f(U)(C) \overset{?}{=} \{y \in QC \mid \text{for all } x \in f_C^{-1}\{y\},\, x \in UC\}\,.$$

is that it is not functorial in C ! In order to correct this, have to modify it by taking instead

$$\forall_f(U)(C) = \{y \in QC \mid \text{for all } h : D \to C, \text{for all } x \in f_D^{-1}\{y.h\},\, x \in UD\}\,, \qquad (3.35)$$

where we have written $y.h$ for the action of Q on $y \in QC$, i.e. $Q(h)(y) \in QD$.

Lemma 3.3.28. *The specification* (3.35) *is the universal quantifier* $\forall_f$ *in presheaves.*

Proof. Consider the diagram

$$\begin{tikzcd}
\mathsf{y}C \times_Q P \arrow[rr, dotted] \arrow[dr, "y'"] \arrow[ddd] & & U \arrow[dl, tail] \\
& P \arrow[d, "f"] & \\
& Q & \\
\mathsf{y}C \arrow[ur, "y"] \arrow[rr, dotted] & & \forall_f U \arrow[ul, tail]
\end{tikzcd}$$

For all $y \in QC$, we have $y \in \forall_f U$ iff the pullback $y' = f^*y$ factors through $U \rightarrowtail P$, as indicated. Replacing the pullback $\mathsf{y}C \times_Q P$ by its generalized elements, the latter condition is equivalent to saying that for all $\mathsf{y}D$ and $\mathsf{y}h : \mathsf{y}D \to \mathsf{y}C$ and $x \in PD$, if $f \circ x = y \circ \mathsf{y}h$, then $x \in UD$, as shown below.

$$\begin{tikzcd}
\mathsf{y}D \arrow[rrrd, dotted, bend left] \arrow[rrdd, bend left, "x"] \arrow[rdddd, bend right, "\mathsf{y}h"'] & & & \\
& \mathsf{y}C \times_Q P \arrow[rr, dotted] \arrow[dr, "y'"'] \arrow[ddd] & & U \arrow[dl, tail] \\
& & P \arrow[d, "f"] & \\
& & Q & \\
& \mathsf{y}C \arrow[ur, "y"'] \arrow[rr, dotted] & & \forall_f U \arrow[ul, tail]
\end{tikzcd}$$

But the last condition is equivalent to saying for all D and all $h : D \to C$ and all $x \in PD$, if $x \in f_D^{-1}\{y.h\}$, then $x \in UD$, which is the righthand side of (3.35). □

Proposition 3.3.29. *For any natural transformation $f : P \to Q$, there are adjoints*

$$\begin{tikzcd}
\mathsf{Sub}(P) \arrow[r, bend left, "\exists_f"] \arrow[r, bend right, "\forall_f"'] & \mathsf{Sub}(Q) \arrow[l, "f^*" description]
\end{tikzcd} \qquad \exists_f \dashv f^* \dashv \forall_f \,.$$

These are determined by the following formulas, where $U \rightarrowtail P$ and $V \rightarrowtail Q$ and $C \in \mathbb{C}$:

$$\begin{aligned}
f^*(V)(C) &= \{x \in PC \mid f_C(x) \in VC\} \qquad (3.36)\\
\exists_f(U)(C) &= \{y \in QC \mid \text{for some } x \in PC,\ f_C(x) = y \ \&\ x \in UC\}\\
\forall_f(U)(C) &= \{y \in QC \mid \text{for all } h : D \to C, \text{ for all } x \in PD,\ f_D(x) = y.h \text{ implies } x \in UD\}
\end{aligned}$$

□

The implication $U \Rightarrow V$ for $U, V \in \mathsf{Sub}(P)$ therefore has the form, for each $C \in \mathbb{C}$,

$$(U \Rightarrow V)(C) \;=\; \{x \in PC \mid \text{for all } h : D \to C,\ x.h \in UD \text{ implies } x.h \in VD\}\,.$$

And the negation $\neg U \in \mathsf{Sub}(P)$ is then, for each $C \in \mathbb{C}$,

$$(\neg U)(C) \;=\; \{x \in PC \mid \text{for all } h : D \to C,\ x.h \notin UD\}\,.$$

Exercise 3.3.30. Prove the last two statements, computing $U \Rightarrow V$ and $\neg U$.

Sets through time. For presheaves on a poset K, the foregoing description of the Heyting structure becomes a bit simpler. Let us consider "covariant presheaves", i.e. functors $A : K \to \mathsf{Set}$. We can regard such a functor as a "set developing through (branching) time", with each later time $i \leq j$ giving rise to a transition map $A_i \to A_j$, which we may denote by

$$A_i \ni a \longmapsto a_j \in A_j \,.$$

For any map $f : A \to B$ (a family of functions $f_i : A_i \to B_i$ compatible with the development over time), we again have the adjunctions

$$\begin{tikzcd} \mathsf{Sub}(A) \arrow[r, bend left, "\exists_f"] \arrow[r, bend right, "\forall_f"'] & \mathsf{Sub}(B) \arrow[l, "f^*"'] \end{tikzcd} \qquad \exists_f \dashv f^* \dashv \forall_f \,.$$

These can now be described by the following formulas, where $U \in \mathsf{Sub}(A)$ and $V \in \mathsf{Sub}(B)$ and $i \in K$:

$$\begin{aligned} f^*(V)_i &= \{x \in A_i \mid f_i(x) \in V_i\} \\ \exists_f(U)_i &= \{y \in B_i \mid \text{for some } x \in A_i,\ f_i(x) = y \ \&\ x \in U_i\} \\ \forall_f(U)_i &= \{y \in B_i \mid \text{for all } j \geq i, \text{ for all } x \in A_j,\ f_j(x) = y_j \text{ implies } x \in U_j\} \end{aligned} \tag{3.37}$$

The implication $U \Rightarrow V$ for $U, V \in \mathsf{Sub}(A)$ then has the form, for each $i \in K$,

$$(U \Rightarrow V)_i \ = \ \{x \in A_i \mid \text{for all } j \geq i,\ x_j \in U_j \text{ implies } x_j \in V_j\} \,.$$

And the negation $\neg U \in \mathsf{Sub}(A)$ is then, for each $i \in K$,

$$(\neg U)_i \ = \ \{x \in A_i \mid \text{for all } j \geq i,\ x_j \notin U_j\} \,.$$

Exercise 3.3.31. Show that for the arrow category $\mathbf{2} = \cdot \to \cdot$ the functor category $\mathsf{Set}^{\to}$ is *not* Boolean.

Remark 3.3.32 (Bi-Heyting categories)**.** We know by Proposition 3.3.29 that in presheaf categories $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, each subobject lattice $\mathsf{Sub}(P)$ is a Heyting algebra. Define a *bi-Heyting category* to be a Heyting category in which each $\mathsf{Sub}(P)$ is a *bi-Heyting algebra*, meaning that both $\mathsf{Sub}(P)$ and its opposite $\mathsf{Sub}(P)^{\mathsf{op}}$ are Heyting algebras. One can show that any presheaf category is also bi-Heyting (this follows from the fact that limits and colimits in presheaves are computed pointwise, but see also Exercise 3.3.33 below). See [Law91, MR95, GER96] for more on bi-Heyting categories.

Exercise 3.3.33. Complete the following sketch to show that any presheaf category $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ is bi-Heyting.

1. Every presheaf P is covered by a coproduct of representables,

$$\coprod_{C \in \mathbb{C},\, x \in PC} \mathsf{y}C \twoheadrightarrow P \,.$$

2. There is therefore an injective lattice homomorphism

$$\mathsf{Sub}(P) \rightarrowtail \prod_{C \in \mathbb{C}, x \in PC} \mathsf{Sub}(\mathsf{y}C)\,.$$

3. It thus suffices to show that all $\mathsf{Sub}(\mathsf{y}C)$ are bi-Heyting.

4. The poset $\mathsf{Sub}(\mathsf{y}C)$ is isomorphic to the poset of *sieves* on C in $\mathbb{C}$: sets S of arrows with codomain C, closed under precomposition by arbitrary arrows, i.e. $(s : C' \to C) \in S$ and $t : C'' \to C'$ implies $s \circ t \in S$.

5. Writing $|\mathbb{D}|$ for the poset reflection of an arbitrary category $\mathbb{D}$, the sieves on C are the same as lower sets in the poset reflection of the slice category $|\mathbb{C}/C|$, thus $\mathsf{Sub}(\mathsf{y}C) \cong {\downarrow}|\mathbb{C}/C|$.

6. For *any* poset P, the poset of lower sets ${\downarrow}P$, ordered by inclusion, form a Heyting algebra.

7. The opposite category of ${\downarrow}P$ is isomorphic to the upper sets ${\uparrow}P$.

8. But since ${\uparrow}P = {\downarrow}(P^{\mathsf{op}})$, by (6) the poset $({\downarrow}P)^{\mathsf{op}}$ is also a Heyting algebra.

9. Thus $\mathsf{Sub}(\mathsf{y}C)$ is a bi-Heyting algebra.

Remark 3.3.34 (First-order logical duality)**.** The Stone duality for Boolean algebras was seen in Section 2.7 to have a logical interpretation, under which Boolean algebras represent theories in propositional logic, and Stone spaces represent their $\mathbb{2}$-valued semantics, with valuations as the points of the corresponding Stone space. There is an analogous duality theory for first-order logic, which extends and generalizes both that for propositional logic as well as that for algebraic theories (Lawvere duality 1.2). Theories are represented by Boolean *categories* and their (Set-valued) semantics by topological *groupoids* of models. The interested reader may consult the sources ([Mak93, Mak87], [AF13, Awo21]).

3.3.4 Kripke-Joyal semantics

In section 3.1.2, we introduced the idea of using "generalized elements" $z : Z \to C$ as a way of externalizing the interpretation of the logical language. With respect to a subobject $S \rightarrowtail C$, such an element is said to be *in the subobject*, writtten $z \in_C S$, if it factors through $S \rightarrowtail C$.

$$\begin{tikzcd} & S \arrow[d, tail] \\ Z \arrow[ru, dotted] \arrow[r, "z"'] & C \end{tikzcd}$$

Generalized elements provide a way of *testing for satisfaction* of a formula in context $(x : A \mid \varphi)$ by a model M, as follows. Let A_M be the interpretion of the type A in the model M, so that the formula determines a subobject $[\![x : A \mid \varphi]\!]_M \rightarrowtail A_M$. Note that in Heyting logic, with $\forall$ and $\Rightarrow$, we can consider satisfaction of individual formulas in context $(x : A \mid \varphi)$ rather than entailments $(x : A \mid \varphi \vdash \psi)$, by replacing the latter with the equivalent $(x : A \vdash \varphi \Rightarrow \psi)$ — or even, for that matter, $(\top \vdash \forall x : A.\varphi \Rightarrow \psi)$.

Definition 3.3.35. For a theory $\mathbb{T}$ in first-order logic we say that a model M *satisfies* a formula in context $(x : A \mid \varphi)$, written $M \models (x : A \mid \varphi)$, if the subobject $[\![x : A \mid \varphi]\!]_M \rightarrowtail A_M$ is the maximal one 1_{A_M}.

Note that this notion of satisfaction of a formula agrees with our previous notion of satisfaction for the entailment $x : A \mid \top \vdash \varphi$,

$$\begin{aligned} M \models (x : A \mid \varphi) \quad &\text{iff} \quad [\![x : A \mid \varphi]\!]_M = 1_{A_M} \\ &\text{iff} \quad M \models (x : A \mid \top \vdash \varphi)\,. \end{aligned} \tag{3.38}$$

Now observe that the condition $[\![x : A \mid \varphi]\!]_M = 1_{A_M}$ holds just in case every element $z : Z \to A_M$ factors through the subobject $[\![x : A \mid \varphi]\!]_M \rightarrowtail A_M$. It is convenient to use the *forcing* notation $\Vdash$ for this condition, writing

$$Z \Vdash \varphi(z) \quad \text{for} \quad z \in_{A_M} [\![x : A \mid \varphi]\!]_M\,.$$

We can then use forcing to test for satisfaction, by asking whether *all* generalized elements $z : Z \to A_M$ factor through $[\![x : A \mid \varphi]\!]_M \rightarrowtail A_M$, and thus "force" the formula $(x : A \mid \varphi)$:

$$M \models (x : A \mid \varphi) \qquad \text{iff} \qquad \text{for all } z : Z \to A_M,\ Z \Vdash \varphi(z)\,.$$

We summarize these conventions in the following Definition and Lemma.

Definition 3.3.36 (Kripke-Joyal Forcing)**.** In any Heyting category $\mathcal{C}$, define the *forcing relation* $\Vdash$ as follows: for a formula in context $(x : A \mid \varphi)$ in the langage of a theory $\mathbb{T}$, and a $\mathbb{T}$-model M, let A_M interpret the type symbol A; then for any $z : Z \to A_M$, we define the relation "z forces φ" by

$$\begin{aligned} Z \Vdash \varphi(z) \quad &\text{iff} \quad z \in_{A_M} [\![x : A \mid \varphi]\!]_M \\ &\text{iff} \quad z : Z \to A_M \text{ factors as} \end{aligned} \tag{3.39}$$

$$\begin{tikzcd} & {[\![x : A \mid \varphi]\!]_M\ .} \arrow[d, tail] \\ Z \arrow[ur, dotted] \arrow[r, "z"'] & A_M \end{tikzcd}$$

Lemma 3.3.37. *For any model M, we have:*

$$M \models (x : A \mid \varphi) \qquad \textit{iff} \qquad \textit{for all } z : Z \to A_M,\ Z \Vdash \varphi(z)\,. \tag{3.40}$$

□

Of course, we also define forcing for formulas with a context of variables $\Gamma = x_1 : A_1, \ldots x_n : A_n$, and then we have

$$M \models (\Gamma \mid \varphi) \qquad \text{iff} \qquad \text{for all } z : Z \to \Gamma_M,\ Z \Vdash \varphi(z)\,.$$

where $\Gamma_M = (A_1)_M \times \ldots \times (A_1)_M$, and $\varphi(z) = \varphi(z_1, \ldots, z_n)$ where $z_i = \pi_i z : Z \to \Gamma_M \to (A_i)_M$. In the extremal case, we have a formula $\cdot \mid \varphi$ with no free variables (a *closed formula* or *sentence*), for which the interpretation $[\![\cdot \mid \varphi]\!] \rightarrowtail 1$ is in $\mathsf{Sub}(1)$. For such a closed formula, we have

$$\begin{aligned} M \models (\cdot \mid \varphi) \qquad &\text{iff} \qquad \text{for all } z : Z \to 1,\ Z \Vdash \varphi \\ &\text{iff} \qquad [\![\cdot \mid \varphi]\!] = 1\,. \end{aligned} \tag{3.41}$$

In this sense, the Heyting algebra $\mathsf{Sub}(1)$ contains the *truth-values* of statements $(\cdot \mid \varphi)$ in the internal logic, which hold if and only if $[\![\cdot \mid \varphi]\!] = 1$.

The forcing relation $Z \Vdash \varphi(z)$ defined in (3.39) allows us to turn an internal statement $[\![x : A \mid \varphi]\!]_M$, i.e. a formula interpreted as an object of $\mathcal{C}$, into an external one, i.e. an ordinary statement that makes reference to objects an arrows of $\mathcal{C}$. We first restrict attention to categories of presheaves $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, for the sake of simplicity (but see Remark 3.3.39 below.) In this case, we can restrict to generalized elements $z : Z \to A_M$ of the special form $c : \mathsf{y}C \to A_M$, i.e. with representable domains, because Lemma 3.3.37 clearly still holds when so restricted: $M \models (x : A \mid \varphi)$ iff for all $c : \mathsf{y}C \to A_M$, we have $\mathsf{y}C \Vdash \varphi(c)$. Moreover, we then write simply $C \Vdash \varphi(c)$ for $\mathsf{y}C \Vdash \varphi(c)$. Observe that because (by Yoneda) $c : \mathsf{y}C \to A_M$ corresponds to $c \in A_M(C)$ in Set, with subset $([\![x : A \mid \varphi]\!]_M)(C) \subseteq A_M(C)$, we have, finally, the equivalence

$$C \Vdash \varphi(c) \quad \text{iff} \quad c \in [\![x : A \mid \varphi]\!]_M(C)\,. \tag{3.42}$$

Theorem 3.3.38 (Kripke-Joyal Semantics)**.** *For any presheaf category $\widehat{\mathbb{C}}$ and model M of a theory $\mathbb{T}$ in first-order logic, let $(x : A \mid \varphi)$, $(x : A \mid \psi)$, and $(x : A, y : B \mid \vartheta)$ be formulas (in context) in the language of $\mathbb{T}$, and let $C \in \mathbb{C}$ and $c, c_1, c_2 : \mathsf{y}C \to A_M$ be any maps. Then we have*

1. $C \Vdash \top(c)$ *always.*

2. $C \Vdash \bot(c)$ *never.*

3. $C \Vdash c_1 = c_2 \quad$ *iff* $\quad c_1 = c_2$ *as arrows* $\mathsf{y}C \to A_M$.

4. $C \Vdash \varphi(c) \wedge \psi(c) \quad$ *iff* $\quad C \Vdash \varphi(c)$ *and* $C \Vdash \psi(c)$.

5. $C \Vdash \varphi(c) \vee \psi(c) \quad$ *iff* $\quad C \Vdash \varphi(c)$ *or* $C \Vdash \psi(c)$.

6. $C \Vdash \varphi(c) \Rightarrow \psi(c) \quad$ *iff* $\quad$ *for all* $d : D \to C,\ D \Vdash \varphi(c.d)$ *implies* $D \Vdash \psi(c.d)$.

7. *$C \Vdash \neg\varphi(c)$ iff for no $d : D \to C$, $D \Vdash \varphi(c.d)$.*

8. *$C \Vdash \exists y : B.\, \vartheta(c, y)$ iff for some $c' : C \to B_M$, $C \Vdash \vartheta(c, c')$.*

9. *$C \Vdash \forall y : B.\, \vartheta(c, y)$ iff for all $d : D \to C$, for all $d' : D \to B_M$, $D \Vdash \vartheta(c.d, d')$.*

Proof. We just do a few cases and leave the rest to the reader.

...

Use (3.36) for the non-obvious cases. □

Examples: LEM, DN, a map is epic, monic, iso. Constant domains.

Remark 3.3.39. There are several variations on Kripke-Joyal semantics for various special kinds of categories: presheaves on a poset P, sheaves on a topological space or a complete Heyting algebra, G-sets for a group or groupoid G, sheaves on a Grothendieck site (i.e. a Grothendieck topos), as well as a general case for arbitrary Heyting categories. Many of these are discussed in [MM92]. In the case of sheaves, the clauses for falsehood $\bot$, disjunction $\vee$, and the existential quantifier $\exists$ typically become more involved. The result is then akin to what is known in constructive logic as Beth semantics.

We next consider another case that is even simpler than presheaves, namely covariant Set-valued functors on a poset P, which may be called "Kripke models".

Exercise 3.3.40. Show that for a group G, regarded as a category with one object, the functor category Set^G is Boolean.

Exercise 3.3.41. Prove Lemma 3.3.37 in the restricted case of presheaves and generalized elements with representable domains, $a : \mathsf{y}C \to A_M$.

Kripke models

As already mentioned, we can regard covariant functors $A : K \to \mathsf{Set}$ on a *poset* K as "sets developing through time". A model in such a category Set^K is a parametrized family of models, $(M_i)_{i \in I}$, or a *variable model*, which can be thought of as changing through space or (non-linearly ordered) time, represented by K. The satisfaction of a formula by such a variable structure can be tested by forcing, as a special case of Theorem 3.3.38. The result becomes simplified somewhat in the clauses for $\forall$ and $\Rightarrow$, in a way that agrees with the original semantics of Kripke [**?**].

Theorem 3.3.42 (Kripke Semantics)**.** *For any first-order theory $\mathbb{T}$ and poset K and model M in the functor category Set^K, let $(x : A \mid \varphi)$, $(x : A \mid \psi)$, and $(x : A, y : B \mid \vartheta)$ be formulas in context in the language of $\mathbb{T}$, and let $i \in K$ and $a, a_1, a_2 : \mathsf{y}i \to A_M$ be any maps (respectively elements $a, a_1, a_2 \in (A_M)_i$. Then for each $i \in K$ we write $i \Vdash \varphi(a)$ for the relation $a \in ([\![x : A \mid \varphi]\!]_M)_i$. We can then calculate:*

1. *$i \Vdash \top(a)$ always.*

2. $i \Vdash \bot(a)$ *never.*

3. $i \Vdash a_1 = a_2$ *iff* $a_1 = a_2$ *as elements of the set* $(A_M)_i$*.*

4. $i \Vdash \varphi(a) \wedge \psi(a)$ *iff* $i \Vdash \varphi(a)$ *and* $i \Vdash \psi(a)$*.*

5. $i \Vdash \varphi(a) \vee \psi(a)$ *iff* $i \Vdash \varphi(a)$ *or* $i \Vdash \psi(a)$*.*

6. $i \Vdash \varphi(a) \Rightarrow \psi(a)$ *iff for all* $j \geq i,\ j \Vdash \varphi(a_j)$ *implies* $j \Vdash \psi(a_j)$*.*

7. $i \Vdash \neg\varphi(a)$ *iff for no* $j \geq i,\ j \Vdash \varphi(a_j)$*.*

8. $i \Vdash \exists y : B.\, \vartheta(a, y)$ *iff for some* $b : \mathsf{y}i \to B_M$*,* $i \Vdash \vartheta(a, b)$*.*

9. $i \Vdash \forall y : B.\, \vartheta(a, y)$ *iff for all* $j \geq i$*, for all* $b : \mathsf{y}j \to B_M$*,* $j \Vdash \vartheta(a_j, b)$*.*

Proof. Use (3.37) for the non-obvious cases. □

Examples: LEM, DN, a map is epic, monic, iso. Constant domain, increasing domain, individuals and trans-world identity. Presheaf of real-valued functions on a space is an ordered ring.

3.3.5 Joyal embedding theorem

We know by Theorem 3.3.22 that intuitionstic first-order logic is complete with respect to models in arbitrary Heyting categories, and moreover, that for every theory $\mathbb{T}$, there is a "generic" model, namely the universal one U in the classifying category $\mathcal{C}_{\mathbb{T}}$. The model U is logically generic in the sense that, for any formula in context $(x : A \mid \varphi)$, we have

$$U \models (x : A \mid \varphi) \quad \text{iff} \quad \mathbb{T} \vdash (x : A \mid \varphi)\,.$$

(The symbol $\vdash$ is once again available for provability from a set of formulas, the axioms of $\mathbb{T}$, now that we can restrict attention to single formulas rather than entailments $\varphi \vdash \psi$; see Definition 3.3.35.)

Lemma 3.3.43. *A functor* $F : \mathcal{C} \to \mathcal{D}$ *is said to be* conservative *if it is faithful and reflects isomorphisms. A Heyting functor between Heyting categories is already conservative if it reflects isos; such a functor induces an injective homomorphism on the Heyting algebras* $\mathsf{Sub}(A)$ *for all* $A \in \mathcal{C}$*.*

Proof. Let $F : \mathcal{C} \to \mathcal{D}$ be Heyting and conservative. The induced functor $\mathsf{Sub}(F) : \mathsf{Sub}(A) \to \mathsf{Sub}(FA)$, taking $U \rightarrowtail A$ to $FU \rightarrowtail FA$, is easily seen to preserve the Heyting operations, because F is Heyting. Just as in the category of groups, a homomorphism of Heyting algebras is injective iff it has a trivial kernel $\mathsf{Sub}(F)^{-1}(1)$. Let $U \rightarrowtail A$ be in the kernel, i.e. $FU \rightarrowtail FA$ is iso. Then $U \rightarrowtail A$ is iso since F is conservative. To see that F is faithful consider the equalizer of a parallel pair of maps. □

By the foregoing lemma, in order to show completeness of first-order intuitionistic logic with respect to the Kripke-Joyal semantics of Theorem 3.3.38, it will suffice if we can embed $\mathcal{C}_{\mathbb{T}}$ by a conservative Heyting functor into a functor category $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ for some suitable (small) category $\mathbb{C}$,

$$F : \mathcal{C}_{\mathbb{T}} \rightarrowtail \widehat{\mathbb{C}}\,.$$

For then, if $FU \models (x : A \mid \varphi)$ in $\widehat{\mathbb{C}}$, then $U \models (x : A \mid \varphi)$ in $\mathcal{C}_{\mathbb{T}}$, since

$$\begin{aligned} FU \models (x : A \mid \varphi) &\text{ iff } 1 = [\![x : A \mid \varphi]\!]_{FU} = F([\![x : A \mid \varphi]\!]_U) \\ &\text{ iff } 1 = [\![x : A \mid \varphi]\!]_U \\ &\text{ iff } U \models (x : A \mid \varphi)\,. \end{aligned}$$

Such an embedding suffices, therefore, to prove completeness with respect to models in categories of the form $\widehat{\mathbb{C}}$, for which we have Kripke-Joyal semantics. The following representation theorem from [MR95] is originally due to Joyal.

Theorem 3.3.44 (Joyal)**.** *For any small Heyting category $\mathcal{H}$ there is a small category $\mathbb{M}$ and a conservative Heyting functor*

$$\mathcal{H} \rightarrowtail \mathsf{Set}^{\mathbb{M}}\,. \tag{3.43}$$

The proof of Joyal's theorem is beyond the scope of these notes, but we will mention that the category $\mathbb{M}$ can be taken to be (a subcategory of) the category of regular functors $\mathcal{H} \to \mathsf{Set}$,

$$\mathbb{M} = \mathsf{Reg}(\mathcal{H}, \mathsf{Set}) \hookrightarrow \mathsf{Set}^{\mathcal{H}},$$

where $\mathsf{Reg}(\mathcal{H}, \mathsf{Set})$ is the category of all *regular* (not Heyting!) functors $\mathcal{H} \to \mathsf{Set}$, and can therefore be regarded as a "category of models" of the "underlying regular theory" of the Heyting category $\mathcal{H}$. The embedding (3.43) is then the "double dual" $\mathcal{H} \rightarrowtail \mathsf{Set}^{\mathsf{Reg}(\mathcal{H},\mathsf{Set})}$, obtained by transposing the evaluation

$$\mathsf{Reg}(\mathcal{H}, \mathsf{Set}) \times \mathcal{H} \longrightarrow \mathsf{Set}$$

which takes $R : \mathcal{H} \to \mathsf{Set}$ and $C \in \mathcal{H}$ to $R(C) \in \mathsf{Set}$. Here we have a glimpse of a generalization of Lawvere duality (as well as Stone duality, as emphasized in [MR95]) to regular categories, as developed by Makkai [**?**]. The conservativity of the embedding (3.43) makes use of the Freyd embedding theorem for regular and coherent categories from Section 3.2.6, but the remarkable fact here is that the "double dual" embedding is not just regular, but actually Heyting. Compare the analogous result for the (special case) of propositional logic given in Chapter 2.

Note that, although $\mathbb{M}$ may be a large category, since $\mathcal{H}$ is small, there is a *small* full subcategory $\mathbb{M}' \hookrightarrow \mathbb{M}$ of "models" that is sufficient to make the embedding conservative.

Theorem 3.3.45. *Intuitionistic first-order logic is sound and complete with respect to the Kripke-Joyal semantics of 3.3.38. Specifically, for every theory $\mathbb{T}$, there is a model M in a presheaf category $\widehat{\mathbb{C}}$ with the property that, for every closed formula φ,*

$$\mathbb{T} \vdash \varphi \quad \textit{iff} \quad M \models \varphi \quad \textit{iff} \quad \mathbb{C} \Vdash \varphi\,,$$

where by $\mathbb{C} \Vdash \varphi$ we mean $C \Vdash \varphi$ for all $C \in \mathbb{C}$.

3.3.6 Kripke completeness

Finally, in order to specialize even further to the case of a Kripke model Set^K for a *poset* K, we can use the following "covering theorem".

Theorem 3.3.46 (Diaconescu). *For any small category $\mathbb{C}$ there is a poset K and a conservative Heyting functor*

$$\mathsf{Set}^{\mathbb{C}} \rightarrowtail \mathsf{Set}^K . \tag{3.44}$$

For a sketch of the proof (see [MM92, IX.9] and [MR95, §3] for details), the poset K may be taken to be $\mathsf{String}(\mathbb{C})$, consisting of finite strings of arrows in $\mathbb{C}$,

$$s = (C_n \xrightarrow{s_n} C_{n-1} \longrightarrow \ldots \longrightarrow C_1 \xrightarrow{s_1} C_0)$$

ordered by $t \leq s$ iff t extends s to the left, i.e. $s_i = t_i$ for all s_i in the string s. There is an evident functor

$$\pi : \mathsf{String}(\mathbb{C}) \longrightarrow \mathbb{C}$$

taking $s = (s_0, \ldots, s_n)$ to the "first" object C_n and $t \leq s$ to the evident composite of the extra initial t's. The functor π induces one on the functor categories by precomposition

$$\pi^* : \mathsf{Set}^{\mathbb{C}} \longrightarrow \mathsf{Set}^{\mathsf{String}(\mathbb{C})} .$$

One can show by a direct calculation that π^* is Heyting and that it is conservative, using the fact that π is surjective on both arrows and objects.

Corollary 3.3.47. *Intuitionistic first-order logic is sound and complete with respect to the Kripke semantics of Theorem 3.3.42. Specifically, for every theory $\mathbb{T}$, there is a poset K and a model M in Set^K with the property that, for every closed formula φ,*

$$\mathbb{T} \vdash \varphi \quad \textit{iff} \quad M \models \varphi \quad \textit{iff} \quad K \Vdash \varphi ,$$

where by $K \Vdash \varphi$ we mean $k \Vdash \varphi$ for all $k \in K$.

Remark 3.3.48 (Gödel completeness). Using the fact that a Boolean category is the same thing a coherent category with Boolean subobject lattices, and therefore a Boolean functor between such categories is the same thing as a coherent functor (cf. Lemma **??**), we can specialize the completeness theorem for coherent logic to Boolean categories and Set-valued completeness, i.e., the classical Gödel completeness theorem for first-order logic. This formulation is sometimes called the Gödel-Deligne-Joyal completeness theorem.

3.4 Hyperdoctrines

For a given algebraic signature, let $\mathcal{C}$ be the category of contexts $\Gamma = (x_1 : X_1, ..., x_n : X_n)$ with n-tuples of terms in context $\Delta = (y_1 : Y_1, ..., y_m : Y_m)$ as arrows $\sigma : \Delta \to \Gamma$. Composition is given by substitution, and the identity arrows by variables (terms are identified up to α-renaming of variables, as in the Lawvere theories of Chapter 1). The category $\mathcal{C}$ then has all finite products. For each object Γ, let $P(\Gamma)$ be the poset of all first-order formulas $(\Gamma \mid \varphi)$, up to provable equivalence. Substitution of a term $\sigma : \Delta \to \Gamma$ into a formula $(\Gamma \mid \varphi)$ determines a morphism of posets $\sigma^* : P(\Gamma) \to P(\Delta)$, which also preserves all of the propositional operations,

$$\sigma^*(\varphi \wedge \psi) = \varphi[\sigma/x] \wedge \psi[\sigma/x] = \sigma^*(\varphi) \wedge \sigma^*(\psi),$$

etc. Moreover, since substitutions into formulas and terms commute with each other, $\tau^*\sigma^*\varphi = \varphi[\sigma \circ \tau/x]$, this action is strictly functorial, so we have a contravariant functor

$$P : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathsf{Heyt}$$

from the category of contexts to the category of Heyting algebras.

Now consider the quantifiers $\exists$ and $\forall$. Given a projection of contexts $p_X : \Gamma \times X \to \Gamma$, in addition to the pullback functor

$$p_X^* : P(\Gamma) \longrightarrow P(\Gamma \times X)$$

induced by weakening, there are the operations of quantification

$$\exists_X, \forall_X : P(\Gamma \times X) \longrightarrow P(\Gamma)\,.$$

By the rules for the quantifiers, these are left and right adjoints to weakening,

$$\exists_X \dashv p_X^* \dashv \forall_X\,.$$

The Beck-Chevalley rules are also satisfied, because substitution respects quantifiers, in the sense that $(\forall_x \varphi)[s/y] = \forall_x(\varphi[s/y])$.

Definition 3.4.1. A *(posetal) hyperdoctrine* consists of a Cartesian category $\mathcal{C}$ together with a contravariant functor

$$P : \mathcal{C}^{\mathsf{op}} \longrightarrow \mathsf{Heyt}\,,$$

such that for each $f : D \to C$ the action maps $f^* = Pf : PC \to PD$ have both left and right adjoints

$$\exists_f \dashv f^* \dashv \forall_f$$

that satisfy the Beck-Chavalley conditions.

Examples

1. We already saw the syntactic example of first-order logic. For each first-order theory $\mathbb{T}$ there is an associated hyperdoctrine $(\mathcal{C}_{\mathbb{T}}, P_{\mathbb{T}})$, with the types and terms of $\mathbb{T}$ as the category of contexts $\mathcal{C}_{\mathbb{T}}$, and the formulas (in context) of $\mathbb{T}$ as "predicates", i.e. the elements of the Heyting algebras $\varphi \in P_{\mathbb{T}}(\Gamma)$. A general hyperdoctrine can be regarded as an abstraction of this example.

2. A hyperdoctrine on the index category $\mathcal{C} = \mathsf{Set}$ is given by the powerset functor
$$\mathcal{P} : \mathsf{Set}^{\mathsf{op}} \to \mathsf{Heyt}\,,$$
which is represented by the Heyting algebra $\mathbb{2}$, in the sense that for each set I one has
$$\mathcal{P}I \cong \mathsf{Hom}(I, \mathbb{2})\,.$$
Similarly, for any complete Heyting algebra H, there is a hyperdoctrine H-Set, with
$$P_{\mathsf{H}}(I) \cong \mathsf{Hom}(I, \mathsf{H})\,.$$
The adjoints to precomposition along a map $f : J \to I$ are given by
$$\begin{aligned} \exists_f(\varphi)(i) &= \bigvee_{j \in J} i = f(j) \wedge \varphi(j)\,, \\ \forall_f(\varphi)(i) &= \bigwedge_{j \in J} i = f(j) \Rightarrow \varphi(j)\,, \end{aligned}$$
where the value of $x = y$ in H is $\bigvee\{\top \mid x = y\}$.

We leave it as an exercise to show that the Beck-Chevalley conditions are satisfied.

Exercise 3.4.2. Show this.

3. For a related example, let $\mathbb{C}$ be any small index category and $\mathcal{C} = \widehat{\mathbb{C}}$, the category of presheaves on $\mathbb{C}$. An internal Heyting algebra H in $\mathcal{C}$, i.e. a functor $\mathbb{C}^{\mathsf{op}} \to \mathsf{Heyt}$, is said to be *internally complete* if, for every $I \in \mathcal{C}$, the transpose $\mathsf{H} \to \mathsf{H}^I$ of the projection $\mathsf{H} \times I \to \mathsf{H}$ has both left and right adjoints. Such an internally complete Heyting algebra determines a (representable) hyperdoctrine $P_{\mathsf{H}} : \mathcal{C} \to \mathsf{Set}$ just as for the case of $\mathcal{C} = \mathsf{Set}$, by setting $P_{\mathsf{H}}(C) = \mathcal{C}(C, \mathsf{H})$.

4. For any Heyting category $\mathcal{H}$ let $\mathsf{Sub}(C)$ be the Heyting algebra of all subobjects $S \rightarrowtail C$ of the object C. The presheaf $\mathsf{Sub} : \mathcal{H}^{\mathsf{op}} \to \mathsf{Heyt}$, with action by pullback, is then a hyperdoctrine, essentially by the definition of a Heyting category.

Remark 3.4.3 (Lawvere's Law)**.** In any hyperdoctrine $(\mathcal{C}, P)$, for each object $C \in \mathcal{C}$, an equality relation $=_C$ exists in each $P(C \times C)$, namely
$$(x =_C y) \;=\; \exists_{\Delta_C}(\top)\,,$$
where $\Delta_C : C \to C \times C$ is the diagonal, $\exists_{\Delta_C} \dashv \Delta_C^*$, and $\top \in P(C)$. Displaying variables for clarity, if $\rho(x, y) \in P(C \times C)$ then $\Delta_C^* \rho(x, y) = \rho(x, x) \in PC$ is the contraction of the different variables, and the $\exists_{\Delta_C} \dashv \Delta_C^*$ adjunction can be formulated as the following two-way rule,
$$\frac{x : C \mid \top \vdash \rho(x, x)}{\overline{x : C, y : C \mid (x =_C y) \vdash \rho(x, y)}} \tag{3.45}$$
which expresses that $(x =_C y)$ is the least reflexive relation on C. See [Law70] and Exercise 3.3.17 above.

Exercise 3.4.4. Prove the equivalence of (3.25) and the above hyperdoctrine formulation of Lawvere's Law (3.45).

Proper hyperdoctrines

Now let us consider some hyperdoctrines of a different kind. For any set I, let Set^I be the category of *families of sets* $(A_i)_{i\in I}$, and for $f : J \to I$ let us reindex along f by the precomposition functor $f^* : \mathsf{Set}^I \to \mathsf{Set}^J$, with

$$f^*((A_i)_{i\in I})_j = A_{f(j)}\,.$$

Thus we have a contravariant functor

$$P : \mathsf{Set}^{\mathsf{op}} \to \mathsf{Cat}$$

with $P(I) = \mathsf{Set}^I$ and $f^*(A : I \to \mathsf{Set}) = A \circ f : J \to \mathsf{Set}$.

Lemma 3.4.5. *The precomposition functors $f^* : \mathsf{Set}^I \to \mathsf{Set}^J$ have both left and right adjoints, $f_! \dashv f^* \dashv f_*$, which can be computed by the formulas:*

$$\begin{aligned} f_!(A)_i &= \coprod_{j\in f^{-1}\{i\}} A_j\,, \\ f_*(A)_i &= \prod_{j\in f^{-1}\{i\}} A_j\,, \end{aligned} \tag{3.46}$$

for $A = (A_j)_{j\in J}$. Moreover, these functors satisfy the Beck-Chevally conditions.

Proof. The Beck-Chevalley conditions for such Cat-valued functors are stated as (canonical) isomorphisms, rather than equalities, as they were for poset-valued functors.

□

In this way, the entire hyperdoctrine structure can be weakened to include (coherent) isomorphisms, when the individual categories $P(I)$ are proper categories, and not just posets. We will not specify the required coherences here, but the interested reader may look up the corresponding notion of an *indexed-category*, which is a Cat-valued *pseudofunctor* (see [Joh03, B1.2]).

We conclude this chapter with a few more examples of such proper hyperdoctrines, the "logic" of which generalizes first-order logic, and is better described as dependent type theory.

1. Locally cartesian closed categories. In the previous example, we took $\mathcal{C} = \mathsf{Set}$ and $P : \mathsf{Set}^{\mathsf{op}} \to \mathsf{Cat}$ to be $P(I) = \mathsf{Set}^I$, with action of $f : J \to I$ on $A : I \to \mathsf{Set}$ by precomposition $f^*A = A \circ f : J \to \mathsf{Set}$, which is strictly functorial. There is an equivalent hyperdoctrine with the slice category $\mathsf{Set}/_I$ as the "category of predicates" and action by pullback $f^* : \mathsf{Set}/_I \to \mathsf{Set}/_J$. The equivalence of categories

$$\mathsf{Set}^I \;\simeq\; \mathsf{Set}/_I$$

allows us to use post-composition as the left adjoint $f_! : \mathsf{Set}/_J \to \mathsf{Set}/_I$, rather than the coproduct formula in (3.46). Indeed, this hyperdoctrine structure arises immediately from the locally cartesian closed character of Set. We have the same for any other LCC $\mathcal{E}$, namely the pair $(\mathcal{E}, \mathcal{E}/_{(-)})$ determines a hyperdoctrine, with the action of $\mathcal{E}/_{(-)}$ by pullback, and the left and right adjoints coming from the LCC structure.

Another familiar example related to LCC structure is presheaves on a small category $\mathbb{C}$, where for the slice category $\widehat{\mathbb{C}}/_X$ we have another category of presheaves, namely $\widehat{\int_{\mathbb{C}} X}$, on the category of elements $\int_{\mathbb{C}} X$. For a natural transformation $f : Y \to X$ we have a functor $\int f : \int Y \to \int X$, which induces a triple of adjoints

$$(\textstyle\int f)_! \dashv (\int f)^* \dashv (\int f)_* : \widehat{\int Y} \longrightarrow \widehat{\int X}\,.$$

These satisfy the Beck-Chevalley conditions up to isomorphism, because this indexed category is equivalent to the one coming from the LCC structure,

$$\widehat{\textstyle\int X} \;\simeq\; \widehat{\mathbb{C}}/_X\,,$$

which we know satisfies them.

Note that each of the categories $\widehat{\mathbb{C}}/_X$ is also Cartesian closed and has coproducts $0, X+Y$, so it is a “categorified” Heyting algebra—although we don't make that part of the definition of a hyperdoctrine.

2. For an example not coming from an LCC, consider the category Pos of posets and monotone maps. For each poset K, let us take as the category of predicates $P(K)$ the full subcategory $\mathsf{dFib}(K) \hookrightarrow \mathsf{Pos}/_K$ consisting of the *discrete fibrations*: monotone maps $p : X \to K$ with the “unique lifting property”: for any x and $k \leq p(x)$ there is a unique $x' \leq x$ with $p(x') = k$. Since each category $\mathsf{dFib}(K)$ is equivalent to a category of presheaves $\mathsf{dFib}(K) \simeq \mathsf{Set}^{K^{\mathsf{op}}}$, and pullback along any monotone $f : J \to K$ preserves discrete fibrations, and moreover commutes with the equivalences to the presheaf categories and the precomposition functor $f^* : \hat{K} \to \hat{J}$, we have a hyperdoctrine if only the Beck-Chevalley conditions hold. We leave this as an exercise for the reader.

3. Fibrations of groupoids. Another example of a hyperdoctrine not arising simply from an LCCC is the category Grpd of groupoids and homomorphisms, which is not LCC (cf. [Pal03]). We can however take as the category of predicates $P(G)$ the full subcategory $\mathsf{Fib}(G) \hookrightarrow \mathsf{Grpd}/_G$ consisting of the *fibrations* into G: homomorphisms $p : H \to G$ with the “iso lifting property”: for any $h \in H$ and $\gamma : g \cong p(h)$ there is some $\vartheta : h' \cong h$ with $p(\vartheta) = \gamma$. Now each category $\mathsf{Fib}(G)$ is biequivalent to a category of presheaves of groupoids $\mathsf{Fib}(G) \simeq \mathsf{Grpd}^{G^{\mathsf{op}}}$. It is not so easy to show that this is a (bicategorical) hyperdoctrine; see [HS98].

Exercise 3.4.6. 1. Verify that the pullback of a discrete fibration $X \to K$ along a monotone map $f : J \to K$ exists in Pos, and is again a discrete fibration.

2. Verify the equivalence of categories $\mathsf{dFib}(K) \simeq \mathsf{Set}^{K^{\mathsf{op}}}$.

3. Show the Beck-Chavelley conditions for the indexed category of discrete fibrations of posets.

These examples of proper hyperdoctrines $P : \mathcal{C}^{\mathsf{op}} \to \mathsf{Cat}$ are related to (dependent) type theory in the way that posetal ones $P : \mathcal{C}^{\mathsf{op}} \to \mathsf{Pos}$ are to FOL. There are actually two distinct aspects of this generalization: (1) the individual categories $P(c)$ of values/predicates may be mere posets, or proper categories, (2) the variation over the index category $\mathcal{C}$ of types/contexts (and its adjoints) is accordingly weakened to pseudo-functoriality. We shall consider each of these generalizations in turn in the next chapter on type theory.

Propositional Logic	Simple Type Theory
First-Order Logic	Dependent Type Theory

Chapter 4

Type Theory

4.1 The Curry-Howard correspondence

Consider the following natural deduction proof in propositional calculus.

$$\dfrac{\dfrac{\dfrac{\dfrac{[(A\wedge B)\wedge(A\Rightarrow B)]^1}{A\wedge B}}{A} \qquad \dfrac{[(A\wedge B)\wedge(A\Rightarrow B)]^1}{A\Rightarrow B}}{B}}{(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\;{\scriptstyle (1)}$$

This deduction shows that

$$\vdash (A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B.$$

But so does the following:

$$\dfrac{\dfrac{\dfrac{[(A\wedge B)\wedge(A\Rightarrow B)]^1}{A\Rightarrow B} \qquad \dfrac{\dfrac{[(A\wedge B)\wedge(A\Rightarrow B)]^1}{A\wedge B}}{A}}{B}}{(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\;{\scriptstyle (1)}$$

As does:

$$\dfrac{\dfrac{\dfrac{[(A\wedge B)\wedge(A\Rightarrow B)]^1}{A\wedge B}}{B}}{(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\;{\scriptstyle (1)}$$

There is a sense in which the first two proofs are “equivalent”, but not the first and the third. The relation (or property) of *provability* in propositional calculus $\vdash A$ discards such differences in the proofs that witness it. According to the “proof-relevant” point of view, sometimes called *propositions as types*, one retains as relevant some information about the way in which a proposition is proved. This can be done by annotating the proofs with *proof-terms* as they are constructed, as follows:

$$\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_2(x):A\Rightarrow B} \qquad \dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_1(\pi_1(x)):A}}{\pi_2(x)(\pi_1(\pi_1(x))):B}}{\lambda x.\pi_2(x)(\pi_1(\pi_1(x))):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\,{\scriptstyle(1)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_1(\pi_1(x)):A} \qquad \dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_2(x):A\Rightarrow B}}{\pi_2(x)(\pi_1(\pi_1(x))):B}}{\lambda x.\pi_2(x)(\pi_1(\pi_1(x))):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\,{\scriptstyle(1)}$$

$$\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_2(\pi_1(x)):B}}{\lambda x.\pi_2(\pi_1(x)):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\,{\scriptstyle(1)}$$

The proof terms for the first two proofs are the same, namely $\lambda x.\pi_2(x)(\pi_1(\pi_1(x)))$, but the term for the third one is $\lambda x.\pi_2(\pi_1(x))$, reflecting the difference in the proofs. The assignment works by labelling assumptions as variables, and then associating term-constructors to the different rules of inference: pairing and projection to conjunction introduction and elimination, function application and λ-abstraction to implication elimination (*modus ponens*) and introduction. The use of variable binding to represent cancellation of premisses is a particularly effective device.

From the categorical point of view, the relation of deducibility $A \vdash B$ is a mere preorder. The addition of proof terms $x : A \vdash t : B$ results in a *categorification* of this preorder, in the sense that it becomes a "proper" category, the preordered reflection of which is the deducibility preorder. And now a remarkable fact emerges: it is hardly surprising that the deducibility preorder has, say, finite products $A \wedge B$ or even exponentials $A \Rightarrow B$; but it is *amazing* that the category with proof terms $x : A \vdash t : B$ as arrows also turns out to be a cartesian closed category, and indeed a proper one, with distinct parallel arrows, such as

$$\begin{aligned} \pi_2(x)(\pi_1(\pi_1(x))) : (A\wedge B)\wedge(A\Rightarrow B) &\longrightarrow B, \\ \pi_2(\pi_1(x)) : (A\wedge B)\wedge(A\Rightarrow B) &\longrightarrow B. \end{aligned}$$

This *category of proofs* contains information about the "proof theory" of the propositional calculus, as opposed to its mere relation of deducibility.

And now another remarkable fact emerges: when the calculus of proof terms is formulated as a system of *simple type theory*, it admits an alternate interpretation as a formal system of *function abstraction and application*. This dual interpretation of the system of type theory—as the proof theory of propositional logic, and as formal system for manipulating functions—is sometimes called the *Curry-Howard correspondence* [Sco70, ML84, Tai68]. From the categorical point of view, it expresses a structural equivalence between the cartesian closed categories of *proofs* in propositional logic and *terms* in simple type theory, both of which are categorifications of their common preorder reflection, the deducibility preorder of propositional logic (cf. [MH92]).

In the following sections, we shall consider this remarkable correspondence in detail, as well as some extensions of the basic case represented by cartesian closed categories: categories with coproducts, cocomplete categories, and categories equipped with modal operators. In the next chapter, it will be seen that this correspondence even extends to proofs in quantified predicate logic and terms in dependent type theory, and beyond.

4.2 Cartesian closed categories

Exponentials

We begin with the notion of an exponential B^A of two objects A, B in a category, motivated by a couple of important examples. Consider first the category Pos of posets and monotone functions. For posets P and Q the set $\mathsf{Hom}(P, Q)$ of all monotone functions between them is again a poset, with the pointwise order:

$$f \leq g \iff fx \leq gx \quad \text{for all } x \in P \ . \qquad\qquad (f, g : P \to Q)$$

Thus, when equipped with a suitable order, the *set* $\mathsf{Hom}(P, Q)$ becomes an object of Pos.

Similarly, given monoids $K, M \in \mathsf{Mon}$, there is a natural monoid structure on the set $\mathsf{Hom}(K, M)$, defined pointwise by

$$(f \cdot g)x = fx \cdot gx \ . \qquad\qquad (f, g : K \to M,\ x \in K)$$

Thus the category Mon also admits such "internal Homs". The same thing works in the category Group of groups and group homomophisms, where the set $\mathsf{Hom}(G, H)$ of all homomorphisms between groups G and H can be given a pointwise group structure.

These examples suggest a general notion of an "internal Hom" in a category: an "object of morphisms $A \to B$" which corresponds to the hom-set $\mathsf{Hom}(A, B)$. The other ingredient needed is an "evaluation" operation $\mathsf{eval} : B^A \times A \to B$ which evaluates a morphism $f \in B^A$ at an argument $a \in A$ to give a value $\mathsf{eval} \circ \langle f, a \rangle = f(a) \in B$. This is always going to be present as an operation on underlying sets, if we're starting from a set of functions $\mathsf{Hom}(A, B)$ between structured sets A and B, but even in that case it also needs to be an actual morphism in the category. Finally, we need an operation of "transposition", taking a morphism $f : C \times A \to B$ to one $\widetilde{f} : C \to A^B$. We shall see that this in fact separates the previous two examples.

Definition 4.2.1. In a category $\mathcal{C}$ with binary products, an *exponential* (B^A, ϵ) of objects A and B is an object B^A together with a morphism $\epsilon : B^A \times A \to B$, called the *evaluation* morphism, such that for every $f : C \times A \to B$ there exists a *unique* morphism $\widetilde{f} : C \to B^A$, called the *transpose*[1] of f, for which the following diagram commutes.

$$\begin{tikzcd}
B^A & B^A \times A \arrow[r, "\epsilon"] & B \\
C \arrow[u, dotted, "\widetilde{f}"] & C \times A \arrow[u, "\widetilde{f} \times 1_A"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Commutativity of the diagram of course means that $\epsilon \circ (\widetilde{f} \times 1_A) = f$.

Definition 4.2.1 is called the *universal property of the exponential.* It is just the category-theoretic way of saying that a function $f : C \times A \to B$ of two variables can be viewed as a function $\widetilde{f} : C \to B^A$ of one variable that maps $z \in C$ to a function $\widetilde{f}z = f\langle z, -\rangle : A \to B$ that maps $x \in A$ to $f\langle z, x\rangle$. The relationship between f and $\widetilde{f}$ is then the expected one:

$$(\widetilde{f}z)x = f\langle z, x\rangle \ .$$

That is all there is to it, except that by making the evaluation explicit, variables and elements never need to be mentioned! The benefit of this is that the definition makes sense also in categories whose objects are not *sets*, and whose morphisms are not *functions*—even though some of the basic examples are of that sort.

In Poset the exponential Q^P of posets P and Q is the set of all monotone maps $P \to Q$, ordered pointwise, as above. The evaluation map $\epsilon : Q^P \times P \to Q$ is just the usual evaluation of a function at an argument. The transpose of a monotone map $f : R \times P \to Q$ is the map $\widetilde{f} : R \to Q^P$, defined by, $(\widetilde{f}z)x = f\langle z, x\rangle$, i.e. the transposed *function.* We say that the category Pos *has all exponentials.*

Definition 4.2.2. Suppose $\mathcal{C}$ has all finite products. An object $A \in \mathcal{C}$ is *exponentiable* when the exponential B^A exists for every $B \in \mathcal{C}$ (along with an associated evaluation map $\epsilon : B^A \times A \to B$). We say that $\mathcal{C}$ *has exponentials* if every object is exponentiable. A *cartesian closed category (ccc)* is a category that has all finite products and exponentials.

Example 4.2.3. Consider again the example of the set $\mathsf{Hom}(M, N)$ of homomorphisms between two monoids M, N, equipped with the pointwise monoid structure. To be a monoid homomorphism, the transpose $\widetilde{h} : 1 \to \mathsf{Hom}(M, N)$ of a homomorphism $h : 1 \times M \to N$ would have to take the unit element $u \in 1$ to the unit homomorphism $u : M \to N$, which is the constant function at the unit $u \in N$. Since $1 \times M \cong M$, that would mean that *all* homomorphisms $h : M \to N$ would have the same transpose, namely $\widetilde{h} = u : 1 \to \mathsf{Hom}(M, N)$. So Mon cannot be cartesian closed. The same argument works in the category Group, and in many related ones.

[1] Also, f is called the transpose of $\widetilde{f}$, so that f and $\widetilde{f}$ are each other's transpose.

Exercise 4.2.4. Recall that monoids and groups can be regarded as (1-object) categories, and then their homomorphisms are just functors. So we have full subcategories,

$$\mathsf{Mon} \hookrightarrow \mathsf{Group} \hookrightarrow \mathsf{Cat}\,.$$

Is the category Cat of all (small) categories and functors cartesian closed? What about the subcategory of all *groupoids*,

$$\mathsf{Grpd} \hookrightarrow \mathsf{Cat}\,,$$

defined as those categories in which every arrow is an iso?

Two characterizations of CCCs

Proposition 4.2.5. *In a category $\mathcal{C}$ with binary products an object A is exponentiable if, and only if, the functor*

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

has a right adjoint

$$-^A : \mathcal{C} \to \mathcal{C}\ .$$

Proof. If such a right adjoint exists then the exponential of A and B is (B^A, ϵ_B), where $\epsilon_B : B^A \times A \to A$ is the counit of the adjunction at B. Indeed, the universal property of the exponential is just the universal property of the counit $\epsilon : (-)^A \Rightarrow 1_{\mathcal{C}}$.

Conversely, suppose for every B there is an exponential (B^A, ϵ_B). As the object part of the right adjoint we then take B^A. For the morphism part, given $g : B \to C$, we can define $g^A : B^A \to C^A$ to be the transpose of $g \circ \epsilon_B$,

$$g^A = (g \circ \epsilon_B)^\sim$$

as indicated below.

$$\begin{tikzcd} B^A \times A \arrow[r, "\epsilon_B"] \arrow[d, "g^A \times 1_A"'] & B \arrow[d, "g"] \\ C^A \times A \arrow[r, "\epsilon_C"'] & C \end{tikzcd} \tag{4.1}$$

The counit $\epsilon : -^A \times A \Longrightarrow 1_{\mathcal{C}}$ at B is then ϵ_B itself, and the naturality square for ϵ is then exactly (4.1), i.e. the defining property of $(f \circ \epsilon_B)^\sim$:

$$\epsilon_C \circ (g^A \times 1_A) = \epsilon_C \circ ((g \circ \epsilon_B)^\sim \times 1_A) = g \circ \epsilon_B\ .$$

The universal property of the counit ϵ is precisely the universal property of the exponential (B^A, ϵ_B) □

Note that because exponentials can be expressed as right adjoints to binary products, they are determined uniquely up to isomorphism. Moreover, the definition of a cartesian closed category can then be phrased entirely in terms of adjoint functors: we just need to require the existence of the terminal object, binary products, and exponentials.

Proposition 4.2.6. *A category $\mathcal{C}$ is cartesian closed if, and only if, the following functors have* right *adjoints:*

$$\begin{aligned} !_{\mathcal{C}} &: \mathcal{C} \to \mathsf{1} \ , \\ \Delta &: \mathcal{C} \to \mathcal{C} \times \mathcal{C} \ , \\ (- \times A) &: \mathcal{C} \to \mathcal{C} \ . \end{aligned} \qquad (A \in \mathcal{C})$$

Here $!_{\mathcal{C}}$ is the unique functor from $\mathcal{C}$ to the terminal category $\mathsf{1}$ and Δ is the diagonal functor $\Delta A = \langle A, A \rangle$, and the right adjoint of $- \times A$ is exponentiation by A.

□

The significance of the adjoint formulation is that it implies the possibility of a purely *equational* specification (adjoint structure on a category is "algebraic", in a sense that can be made precise; see [**?**]). It follows that there is a equational formulation of the definition of a cartesian closed category.

Proposition 4.2.7 (Equational version of CCC)**.** *A category $\mathcal{C}$ is cartesian closed if, and only if, it has the following structure:*

1. *An object $\mathsf{1} \in \mathcal{C}$ and a morphism $!_A : A \to \mathsf{1}$ for every $A \in \mathcal{C}$.*
2. *An object $A \times B$ for all $A, B \in \mathcal{C}$ together with morphisms $\pi_0 : A \times B \to A$ and $\pi_1 : A \times B \to B$, and for every pair of morphisms $f : C \to A$, $g : C \to B$ a morphism $\langle f, g \rangle : C \to A \times B$.*
3. *An object B^A for all $A, B \in \mathcal{C}$ together with a morphism $\epsilon : B^A \times A \to B$, and a morphism $\widetilde{f} : C \to B^A$ for every morphism $f : C \times A \to B$.*

These new objects and morphisms are required to satisfy the following equations:

1. *For every $f : A \to \mathsf{1}$,*
$$f = \, !_A \ .$$
2. *For all $f : C \to A$, $g : C \to B$, $h : C \to A \times B$,*
$$\pi_0 \circ \langle f, g \rangle = f \ , \qquad \pi_1 \circ \langle f, g \rangle = g \ , \qquad \langle \pi_0 \circ h, \pi_1 \circ h \rangle = h \ .$$
3. *For all $f : C \times A \to B$, $g : C \to B^A$,*
$$\epsilon \circ (\widetilde{f} \times 1_A) = f \ , \qquad (\epsilon \circ (g \times 1_A))^{\sim} = g \ .$$
where for $e : E \to E'$ and $f : F \to F'$ we define
$$e \times f := \langle e\pi_0, f\pi_1 \rangle : E \times F \to E' \times F'.$$

These equations ensure that certain diagrams commute and that the morphisms that are required to exist are unique. For example, let us prove that $(A \times B, \pi_0, \pi_1)$ *is the product of* A *and* B*. For* $f : C \to A$ *and* $g : C \to B$ *there exists a morphism* $\langle f, g\rangle : C \to A \times B$*. Equations*

$$\pi_0 \circ \langle f, g\rangle = f \qquad \text{and} \qquad \pi_1 \circ \langle f, g\rangle = g$$

enforce the commutativity of the two triangles in the following diagram:

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

Suppose $h : C \to A \times B$ *is another morphism such that* $f = \pi_0 \circ h$ *and* $g = \pi_1 \circ h$*. Then by the third equation for products we get*

$$h = \langle \pi_0 \circ h, \pi_1 \circ h\rangle = \langle f, g\rangle \ ,$$

and so $\langle f, g\rangle$ *is unique.*

Exercise 4.2.8. Use the equational characterization of CCCs, Proposition 4.2.7, to show that the category Pos of posets and monotone functions *is* cartesian closed, as claimed. Also verify that that Mon is not. Which parts of the definition fail in Mon?

Exercise 4.2.9. Use the equational characterization of CCCs, Proposition 4.2.7, to show that the product category $\Pi_{i \in I}\,\mathcal{C}_i$ of any (set-indexed) family $(\mathcal{C}_i)_{i \in I}$ of cartesian closed categories $\mathcal{C}_i$ is cartesian closed. Is the same true for an arbitrary limit in Cat?

Some proper CCCs

We next review some important examples of (non-poset) cartesian closed categories, most of which have already been discussed.

Example 4.2.10. The first example is the category Set. We already know that the terminal object is a singleton set and that binary products are cartesian products. The exponential of X and Y in Set is just the set of all functions from X to Y,

$$Y^X = \big\{ f \subseteq X \times Y \;\big|\; \forall\, x : X \,.\, \exists!\, y : Y \,.\, \langle x, y\rangle \in f \big\} \ .$$

The evaluation morphism $\mathsf{eval} : Y^X \times X \to Y$ is the usual evaluation of a function at an argument, i.e., $\mathsf{eval}\langle f, x\rangle$ is the unique $y \in Y$ for which $\langle x, y\rangle \in f$.

Example 4.2.11. The category Cat of all small categories is cartesian closed. The exponential of small categories $\mathcal{C}$ and $\mathcal{D}$ is the category $\mathcal{D}^{\mathcal{C}}$ of functors, with natural transformations as arrows (see A.6). Note that if $\mathcal{D}$ is a groupoid (all arrows are isos), then so is $\mathcal{D}^{\mathcal{C}}$. It follows that the category of groupoids is full (even as a 2-category) in Cat. Since limits of groupoids in Cat are also groupoids, the inclusion of the full subcategory $\mathsf{Grpd} \hookrightarrow \mathsf{Cat}$ preserves limits. It also preserves the CCC structure.

Example 4.2.12. The same reasoning as in the previous example shows that the full subcategory $\mathsf{Pos} \hookrightarrow \mathsf{Cat}$ of all small posets and monotone maps is also cartesian closed, and the (limit preserving) inclusion $\mathsf{Pos} \hookrightarrow \mathsf{Cat}$ also preserves exponentials. Note that the (non-full) forgetful functor $U : \mathsf{Pos} \to \mathsf{Set}$ does not, and that $U(Q^P) \subseteq (UQ)^{UP}$ is in general a *proper* subset.

Exercise 4.2.13. There is a full and faithful functor $I : \mathsf{Set} \to \mathsf{Poset}$ that preserves finite limits as well as exponentials. How is this related to the example $\mathsf{Grpd} \hookrightarrow \mathsf{Cat}$?

The foregoing examples are instances of the following general situation.

Proposition 4.2.14. *Let $\mathcal{E}$ be a CCC and $i : \mathcal{S} \hookrightarrow \mathcal{E}$ a full subcategory with finite products and a left adjoint reflection $L : \mathcal{E} \to \mathcal{S}$ preserving finite products. Suppose moreover that for any two objects A, B in $\mathcal{S}$, the exponential iB^{iA} is again in $\mathcal{S}$. Then $\mathcal{S}$ has all exponentials, and these are preserved by i.*

Proof. By assumption, we have $L \dashv i$ with isomorphic counit $LiS \cong S$ for all $S \in \mathcal{S}$. Let us identify $\mathcal{S}$ with the subcategory of $\mathcal{E}$ that is its image under $i : \mathcal{S} \hookrightarrow \mathcal{E}$. The assumption that B^A is again in $\mathcal{S}$ for all $A, B \in \mathcal{S}$, along with the fullness of $\mathcal{S}$ in $\mathcal{E}$, gives the exponentials, and the closure of $\mathcal{S}$ under finite products in $\mathcal{E}$ ensures that the required transposes will also be in $\mathcal{S}$.

Alternately, for any $A, B \in \mathcal{S}$ set $B^A = L(iB^{iA})$. Then for any $C \in \mathcal{S}$, we have natural isos:

$$\begin{aligned}
\mathcal{S}(C \times A, B) &\cong \mathcal{E}(i(C \times A), iB) \\
&\cong \mathcal{E}(iC \times iA, iB) \\
&\cong \mathcal{E}\big(iC, iB^{iA}\big) \\
&\cong \mathcal{E}\big(iC, iL(iB^{iA})\big) \\
&\cong \mathcal{S}\big(C, L(iB^{iA})\big) \\
&\cong \mathcal{S}\big(C, B^A\big)
\end{aligned}$$

where in the fifth line we used the assumption that iB^{iA} is again in $\mathcal{S}$, in the form $iB^{iA} \cong iE$ for some $E \in \mathcal{S}$, which is then necessarily $L(iB^{iA}) = LiE \cong E$. □

A related general situation that covers some (but not all) of the above examples is this:

Proposition 4.2.15. *Let $\mathcal{E}$ be a CCC and $i : \mathcal{S} \hookrightarrow \mathcal{E}$ a full subcategory with finite products and a right adjoint reflection $R : \mathcal{E} \to \mathcal{S}$. If i preserves finite products, then $\mathcal{S}$ also has all exponentials, and these are computed first in $\mathcal{E}$, and then reflected by R into $\mathcal{S}$.*

Proof. For any $A, B \in \mathcal{S}$ set $B^A = R(iB^{iA})$ as described. Now for any $C \in \mathcal{S}$, we have natural isos:

$$\begin{aligned} \mathcal{S}(C \times A, B) &\cong \mathcal{E}(i(C \times A), iB) \\ &\cong \mathcal{E}(iC \times iA, iB) \\ &\cong \mathcal{E}\big(iC, iB^{iA}\big) \\ &\cong \mathcal{S}\big(C, R(iB^{iA})\big) \\ &\cong \mathcal{S}\big(C, B^A\big)\,. \end{aligned}$$

□

An example of the foregoing is the inclusion of the opens into the powerset of points of a space X,

$$\mathcal{O}X \hookrightarrow \mathcal{P}X$$

This frame homomorphism is associated to the map $|X| \to X$ of locales (or in this case, spaces) from the discrete space on the set of points of X.

Exercise 4.2.16. Which of the examples follows from which proposition?

Example 4.2.17. For any set X, the slice category $\mathsf{Set}/_X$ is cartesian closed. The product of $f : A \to X$ and $g : B \to X$ is the pullback $A \times_X B \to X$, which can be constructed as the set of pairs

$$A \times_X B \to X = \{\langle a, b\rangle \mid fa = gb\}\,.$$

The exponential, however, is *not* simply the set

$$\{h : A \to B \mid f = g \circ h\}\,,$$

(what would the projection to X be?), but rather the set of all pairs

$$\{\langle x, h : A_x \to B_x\rangle \mid x \in X,\ f = g \circ h\}\,,$$

where $A_x = f^{-1}\{x\}$ and $B_x = g^{-1}\{x\}$, with the evident projection to X.

Exercise 4.2.18. Prove that $\mathsf{Set}/_X$ is always cartesian closed.

Example 4.2.19. A presheaf category $\widehat{\mathbb{C}}$ is cartesian closed, provided the index category $\mathbb{C}$ is small. To see what the exponential of presheaves P and Q ought to be, we use the Yoneda Lemma. If Q^P exists, then by Yoneda Lemma and the adjunction $(- \times P) \dashv (-^P)$, we have for all $A \in \mathbb{C}$,

$$Q^P(A) \cong \mathsf{Nat}(\mathsf{y}A, Q^P) \cong \mathsf{Nat}(\mathsf{y}A \times P, Q)\,.$$

Because $\mathcal{C}$ is small $\mathsf{Nat}(\mathsf{y}A \times P, Q)$ is a set, so we can *define* Q^P to be the presheaf

$$Q^P = \mathsf{Nat}(\mathsf{y}- \times P, Q)\,.$$

The evaluation morphism $E : Q^P \times P \Longrightarrow Q$ is the natural transformation whose component at A is

$$\begin{aligned} &E_A : \mathsf{Nat}(\mathsf{y}A \times P, Q) \times PA \to QA\ , \\ &E_A : \langle \eta, x \rangle \mapsto \eta_A \langle 1_A, x \rangle\ . \end{aligned}$$

The transpose of a natural transformation $\phi : R \times P \Longrightarrow Q$ is the natural transformation $\widetilde{\phi} : R \Longrightarrow Q^P$ whose component at A is the function that maps $z \in RA$ to the natural transformation $\widetilde{\phi}_A z : \mathsf{y}A \times P \Longrightarrow Q$, whose component at $B \in \mathcal{C}$ is

$$\begin{aligned} &(\widetilde{\phi}_A z)_B : \mathcal{C}(B, A) \times PB \to QB\ , \\ &(\widetilde{\phi}_A z)_B : \langle f, y \rangle \mapsto \phi_B \langle (Rf)z, y \rangle\ . \end{aligned}$$

Exercise 4.2.20. Verify that the above definition of Q^P really gives an exponential of presheaves P and Q.

It follows immediately that the category of graphs Graph is cartesian closed because it is the presheaf category $\mathsf{Set}^{\cdot \rightrightarrows \cdot}$. The same is of course true for the "category of functions", i.e. the arrow category $\mathsf{Set}^{\rightarrow}$, as well as the category of simplicial sets $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ from topology.

Exercise 4.2.21. This exercise is for students with some background in linear algebra. Let Vec be the category of real vector spaces and linear maps between them. Given vector spaces X and Y, the linear maps $\mathcal{L}(X, Y)$ between them form a vector space. So define $\mathcal{L}(X, -) : \mathsf{Vec} \to \mathsf{Vec}$ to be the functor which maps a vector space Y to the vector space $\mathcal{L}(X, Y)$, and it maps a linear map $f : Y \to Z$ to the linear map $\mathcal{L}(X, f) : \mathcal{L}(X, Y) \to \mathcal{L}(X, Z)$ defined by $h \mapsto f \circ h$. Show that $\mathcal{L}(X, -)$ has a left adjoint $- \otimes X$, but also show that this adjoint is *not* the binary product in Vec.

A few other instructive examples that can be explored by the interested reader are the following.

- Etale spaces over a base space X. This category can be described as consisting of *local homeomorphisms* $f : Y \to X$ and commutative triangles over X between such maps. It is equivalent to the category $\mathsf{Sh}(X)$ of *sheaves* on X. See [**?**, ch.n].
- Various subcategories of topological spaces (sequential spaces, compactly-generated spaces). Cf. [**?**].
- Dana Scott's category Equ of equilogical spaces [**?**].

4.3 Simple type theory

The λ-calculus is an abstract theory of functions, much like group theory is an abstract theory of symmetries. There are two basic operations that can be performed with functions. The first one is the *application* of a function to an argument: if f is a function and a is an argument, then fa is the application of f to a, also called the *value* of f at a. The second operation is *abstraction*: if x is a variable and t is an expression in which x may appear, then there is a function f defined by the equation

$$fx = t \; .$$

Here we gave the name f to the newly formed function. But we could have expressed the same function without giving it a name; this is usually written as

$$x \mapsto t \; ,$$

and it means "x is mapped to t". In λ-calculus we use a different notation, which is more convenient when such abstractions are nested, namely

$$\lambda x. \, t \; .$$

This operation is called λ-*abstraction*. For example, $\lambda x. \, \lambda y. \, (x + y)$ is the function that maps an argument a to the function $\lambda y. \, (a + y)$, which maps an argument b to the value $a + b$. The variable x is said to be *bound in* t in the expression $\lambda x. \, t$.

It may seem strange that in specifying the abstraction of a function, we switched from talking about objects (functions, arguments, values) to talking about *expressions*: variables, names, equations. This "syntactic" point of view seems to have been part of the notion of a function from the start, in the theory of algebraic equations. It is the reason that the λ-calculus is part of *logic*, unlike the theory of cartesian closed categories, which remains thoroughly semantical (and "variable-free"). The relation between the two different points of view occupies the rest of this chapter—and, indeed, the entire subject of logic!

There are two kinds of λ-calculus: the *typed* and the *untyped*. In the untyped version there are no restrictions on how application is formed, so that an expression such as

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

is valid, whatever it may mean. We will concentrate here on the typed λ-calculus. In typed λ-calculus every expression has a *type*, and there are rules for forming valid expressions and types. For example, we can only form an application fa when a has a type A and f has a type $A \to B$, which indicates a function taking arguments of type A and giving results of type B. The judgment that expression t has a type A is written as

$$t : A \; .$$

To computer scientists the idea of expressions having types is familiar from programming languages, whereas mathematicians can think of types as sets and read $t : A$ as $t \in A$.

Simply-typed λ-calculus. We now give a more formal definition of what constitutes a *simply-typed* λ*-calculus*. First, we are given a set of *simple types*, which are generated from *basic types* by formation of products and function types:

$$\begin{aligned} \text{Basic types} \quad & \mathtt{B} ::= \mathtt{B}_0 \mid \mathtt{B}_1 \mid \mathtt{B}_2 \cdots \\ \text{Simple types} \quad & A ::= \mathtt{B} \mid A_1 \times A_2 \mid A_1 \to A_2 . \end{aligned}$$

We adopt the convention that function types associate to the right:

$$A \to B \to C \;=\; A \to (B \to C) \;.$$

We assume there is a countable set of *variables* x, y, u, ... We are also given a set of *basic constants*. The set of *terms* is generated from variables and basic constants by the following grammar:

$$\begin{aligned} \text{Variables} \quad & v ::= x \mid y \mid z \mid \cdots \\ \text{Constants} \quad & c ::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\ \text{Terms} \quad & t ::= v \mid c \mid * \mid \langle t_1, t_2 \rangle \mid \mathtt{fst}\, t \mid \mathtt{snd}\, t \mid t_1\, t_2 \mid \lambda x : A \,.\, t \end{aligned}$$

In words, this means:

1. a variable is a term,
2. each basic constant is a term,
3. the constant $*$ is a term, called the *unit*,
4. if u and t are terms then $\langle u, t \rangle$ is a term, called a *pair*,
5. if t is a term then $\mathtt{fst}\, t$ and $\mathtt{snd}\, t$ are terms,
6. if u and t are terms then $u\, t$ is a term, called an *application*
7. if x is a variable, A is a type, and t is a term, then $\lambda x : A \,.\, t$ is a term, called a λ*-abstraction*.

The variable x is *bound* in $\lambda x : A \,.\, t$. Application associates to the left, thus $s\, t\, u = (s\, t)\, u$. The set of *free variables* $\mathsf{FV}(t)$ of a term t is determined as follows:

$$\begin{aligned} \mathsf{FV}(x) &= \{x\} \quad \text{if } x \text{ is a variable} \\ \mathsf{FV}(a) &= \emptyset \qquad \text{if } a \text{ is a basic constant} \\ \mathsf{FV}(\langle u, t \rangle) &= \mathsf{FV}(u) \cup \mathsf{FV}(t) \\ \mathsf{FV}(\mathtt{fst}\, t) &= \mathsf{FV}(t) \\ \mathsf{FV}(\mathtt{snd}\, t) &= \mathsf{FV}(t) \\ \mathsf{FV}(u\, t) &= \mathsf{FV}(u) \cup \mathsf{FV}(t) \\ \mathsf{FV}(\lambda x.\, t) &= \mathsf{FV}(t) \setminus \{x\} \;\;. \end{aligned}$$

If x_1, …, x_n are *distinct* variables and A_1, …, A_n are types then the sequence

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

is a *typing context*, or just *context*. The empty sequence is sometimes denoted by a dot $\cdot$, and it is a valid context. Contexts are denoted by capital Greek letters Γ, Δ, …

A *typing judgment* is a judgment of the form

$$\Gamma \mid t : A$$

where Γ is a context, t is a term, and A is a type. In addition the free variables of t must occur in Γ, but Γ may contain other variables as well. We read the above judgment as "in context Γ the term t has type A". Next we describe the rules for deriving typing judgments.

- Each basic constant $\mathtt{c}_i$ has a uniquely determined type C_i (not necessarily basic):

$$\frac{}{\Gamma \mid \mathtt{c}_i : C_i}$$

- The type of a variable is determined by the context:

$$\frac{}{x_1 : A_1, \ldots, x_i : A_i, \ldots, x_n : A_n \mid x_i : A_i}\ (1 \leq i \leq n)$$

- The constant $*$ has type $\mathtt{1}$:

$$\frac{}{\Gamma \mid * : \mathtt{1}}$$

- The typing rules for pairs and projections are:

$$\frac{\Gamma \mid a : A \qquad \Gamma \mid b : B}{\Gamma \mid \langle a, b \rangle : A \times B} \qquad \frac{\Gamma \mid t : A \times B}{\Gamma \mid \mathtt{fst}\, t : A} \qquad \frac{\Gamma \mid c : A \times B}{\Gamma \mid \mathtt{snd}\, t : B}$$

- The typing rules for application and λ-abstraction are:

$$\frac{\Gamma \mid t : A \to B \qquad \Gamma \mid a : A}{\Gamma \mid t\, a : B} \qquad \frac{\Gamma, x : A \mid t : B}{\Gamma \mid (\lambda x : A \,.\, t) : A \to B}$$

Lastly, we have *equations* between terms: for terms of type A in context Γ,

$$\Gamma \mid s : A\,, \qquad \Gamma \mid t : A\,,$$

the judgment that they are equal is written as

$$\Gamma \mid s = t : A\,.$$

Note that s and t necessarily have the same type; it does *not* make sense to compare terms of different types. We have the following rules for equations, the effect of which is to make equality between terms into an equivalence relation at each type, and a congruence with respect to all of the operations, just as for algebraic theories:

- Equality is an equivalence relation:

$$\frac{}{\Gamma \mid t = t : A} \qquad \frac{\Gamma \mid s = t : A}{\Gamma \mid t = s : A} \qquad \frac{\Gamma \mid s = t : A \qquad \Gamma \mid t = u : A}{\Gamma \mid s = u : A}$$

- The substitution rule:

$$\frac{\Gamma \mid s = t : A \qquad \Gamma, x : A \mid u = v : B}{\Gamma \mid u[s/x] = v[t/x] : B}$$

- The weakening rule:

$$\frac{\Gamma \mid s = t : A}{\Gamma, x : B \mid s = t : A}$$

- Unit type:

$$\frac{}{\Gamma \mid t = * : \mathtt{1}}$$

- Equations for product types:

$$\frac{\Gamma \mid u = v : A \qquad \Gamma \mid s = t : B}{\Gamma \mid \langle u, s \rangle = \langle v, t \rangle : A \times B}$$

$$\frac{\Gamma \mid s = t : A \times B}{\Gamma \mid \mathtt{fst}\, s = \mathtt{fst}\, t : A} \qquad \frac{\Gamma \mid s = t : A \times B}{\Gamma \mid \mathtt{snd}\, s = \mathtt{snd}\, t : A}$$

$$\frac{}{\Gamma \mid t = \langle \mathtt{fst}\, t, \mathtt{snd}\, t \rangle : A \times B}$$

$$\frac{}{\Gamma \mid \mathtt{fst}\, \langle s, t \rangle = s : A} \qquad \frac{}{\Gamma \mid \mathtt{snd}\, \langle s, t \rangle = t : A}$$

- Equations for function types:

$$\frac{\Gamma \mid s = t : A \to B \qquad \Gamma \mid u = v : A}{\Gamma \mid s\, u = t\, v : B}$$

$$\frac{\Gamma, x : A \mid t = u : B}{\Gamma \mid (\lambda x : A \,.\, t) = (\lambda x : A \,.\, u) : A \to B}$$

$$\frac{}{\Gamma \mid (\lambda x : A \,.\, t) u = t[u/x] : A} \qquad (\beta\text{-rule})$$

$$\frac{}{\Gamma \mid \lambda x : A \,.\, (t\, x) = t : A \to B} \text{ if } x \notin \mathsf{FV}(t) \qquad (\eta\text{-rule})$$

This completes the description of a simply-typed λ-calculus.

Simply-typed λ-theories. Apart from the above rules for equality, which are part of the λ-calculus, we might want to impose additional equations between terms. In this case we speak of a *λ-theory*. Thus, a λ-theory $\mathbb{T}$ is given by a set of basic types and a set of basic constants, called the *signature*, and a set of *equations* of the form

$$\Gamma \mid s = t : A \,.$$

Note that we can always state the equations equivalently in *closed form* simply by λ-abstracting all the variables in the context Γ.

We summarize the preceding definitions.

Definition 4.3.1. A *(simply-typed) signature* S is given by a set of *basic types* $(\mathtt{B}_i)_{i\in I}$ together with a set of *basic (typed) constants* $(\mathtt{c}_j : C_j)_{j\in J}$,

$$S \;=\; \big((\mathtt{B}_i)_{i\in I},\, (\mathtt{c}_j : C_j)_{j\in J}\big)\,.$$

A simply-typed λ-theory $\mathbb{T} = (S, E)$ is a simply-typed signature S together with a set of equations between terms,

$$E = \big(u_k = v_k : A_k\big)_{k\in K}\,.$$

Example 4.3.2. The theory of a group is a simply-typed λ-theory. It has one basic type $\mathtt{G}$ and three basic constants, the unit $\mathtt{e}$, the inverse $\mathtt{i}$, and the group operation $\mathtt{m}$,

$$\mathtt{e} : \mathtt{G}\,, \qquad\qquad \mathtt{i} : \mathtt{G} \to \mathtt{G}\,, \qquad\qquad \mathtt{m} : \mathtt{G}\times\mathtt{G} \to \mathtt{G}\,,$$

with the following familiar equations:

$$\begin{aligned} x : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{e}\rangle = x : \mathtt{G} \\ x : \mathtt{G} &\mid \mathtt{m}\langle \mathtt{e}, x\rangle = x : \mathtt{G} \\ x : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{i}\,x\rangle = \mathtt{e} : \mathtt{G} \\ x : \mathtt{G} &\mid \mathtt{m}\langle \mathtt{i}\,x, x\rangle = \mathtt{e} : \mathtt{G} \\ x : \mathtt{G}, y : \mathtt{G}, z : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{m}\langle y, z\rangle\rangle = \mathtt{m}\langle \mathtt{m}\langle x, y\rangle, z\rangle : \mathtt{G} \end{aligned}$$

Example 4.3.3. More generally, any (Lawvere) algebraic theory $\mathbb{A}$ (as in Chapter 1) determines a λ-theory $\mathbb{A}^\lambda$. There is one basic type $\mathtt{A}$ and for each operation f of arity k there is a basic constant $\mathtt{f} : \mathtt{A}^k \to \mathtt{A}$, where $\mathtt{A}^k$ is the k-fold product $\mathtt{A}\times\cdots\times\mathtt{A}$. It is understood that $\mathtt{A}^0 = \mathtt{1}$. The terms of $\mathbb{A}$ are translated to corresponding terms of $\mathbb{A}^\lambda$ in a straightforward manner. For every axiom $u = v$ of $\mathbb{A}$ there is a corresponding one in $\mathbb{A}^\lambda$,

$$x_1 : \mathtt{A}, \ldots, x_n : \mathtt{A} \mid u = v : \mathtt{A}$$

where $x_1, \ldots, x_n$ are the variables occurring in u and v.

Example 4.3.4. The theory of a directed graph is a simply-typed theory with two basic types, $\mathtt{V}$ for vertices and $\mathtt{E}$ for edges, and two basic constants, source $\mathtt{src}$ and target $\mathtt{trg}$,

$$\mathtt{src} : \mathtt{E} \to \mathtt{V}\,, \qquad\qquad \mathtt{trg} : \mathtt{E} \to \mathtt{V}\,.$$

There are no equations.

Example 4.3.5. The theory of a simplicial set is a simply-typed theory with one basic type $\mathtt{X}_n$ for each natural number n, and the following basic constants, also for each n, and each $0 \leq i \leq n$:

$$\mathtt{d}_i : \mathtt{X}_{n+1} \to \mathtt{X}_n \ , \qquad\qquad \mathtt{s}_i : \mathtt{X}_n \to \mathtt{X}_{n+1} \ .$$

The equations are as follows, for all natural numbers i, j:

$$\begin{aligned} \mathtt{d}_i \mathtt{d}_j &= \mathtt{d}_{j-1} \mathtt{d}_i, \quad \text{if } i < j, \\ \mathtt{s}_i \mathtt{s}_j &= \mathtt{s}_{j+1} \mathtt{s}_i, \quad \text{if } i \leq j, \\ \mathtt{d}_i \mathtt{s}_j &= \begin{cases} \mathtt{s}_{j-1} \mathtt{d}_i, & \text{if } i < j, \\ \mathsf{id}, & \text{if } i = j \text{ or } i = j+1, \\ \mathtt{s}_j \mathtt{d}_{i-1}, & \text{if } i > j+1. \end{cases} \end{aligned}$$

Example 4.3.6. An example of a λ-theory found in the theory of programming languages is the mini-programming language *PCF*. It is a theory in simply-typed λ-calculus with a basic type `nat` for natural numbers, and a basic type `bool` of Boolean values,

$$\text{Basic types} \quad \mathtt{B} ::= \mathtt{nat\ type} \mid \mathtt{bool\ type}.$$

There are basic constants zero `0`, successor `succ`, the Boolean constants `true` and `false`, comparison with zero `iszero`, and for each type A the *conditional* $\mathtt{cond}_A$ and the *fixpoint* operator $\mathtt{fix}_A$. They have the following types:

$$\begin{aligned} \mathtt{0} &: \mathtt{nat} \\ \mathtt{succ} &: \mathtt{nat} \to \mathtt{nat} \\ \mathtt{true} &: \mathtt{bool} \\ \mathtt{false} &: \mathtt{bool} \\ \mathtt{iszero} &: \mathtt{nat} \to \mathtt{bool} \\ \mathtt{cond}_A &: \mathtt{bool} \to A \to A \\ \mathtt{fix}_A &: (A \to A) \to A \end{aligned}$$

The equational axioms of PCF are:

$$\begin{gathered} \cdot \mid \mathtt{iszero}\ \mathtt{0} = \mathtt{true} : \mathtt{bool} \\ x : \mathtt{nat} \mid \mathtt{iszero}\,(\mathtt{succ}\, x) = \mathtt{false} : \mathtt{bool} \\ u : A, t : A \mid \mathtt{cond}_A\, \mathtt{true}\ u\ t = u : A \\ u : A, t : A \mid \mathtt{cond}_A\, \mathtt{false}\ u\ t = t : A \\ t : A \to A \mid \mathtt{fix}_A\ t = t\,(\mathtt{fix}_A\ t) : A \end{gathered}$$

Example 4.3.7 (D.S. Scott)**.** Another example of a λ-theory is the *theory of a reflexive type*. This theory has one basic type `D` and two constants

$$\mathtt{r} : \mathtt{D} \to \mathtt{D} \to \mathtt{D} \qquad\qquad \mathtt{s} : (\mathtt{D} \to \mathtt{D}) \to \mathtt{D}$$

satisfying the equation

$$f : \mathtt{D} \to \mathtt{D} \mid \mathtt{r}\,(\mathtt{s}\,f) = f : \mathtt{D} \to \mathtt{D} \tag{4.2}$$

which says that $\mathtt{s}$ is a section and $\mathtt{r}$ is a retraction, so that the function type $\mathtt{D} \to \mathtt{D}$ is a subspace (even a retract) of $\mathtt{D}$. A type with this property is said to be *reflexive.* We may additionally stipulate the axiom

$$x : \mathtt{D} \mid \mathtt{s}\,(\mathtt{r}\,x) = x : \mathtt{D} \tag{4.3}$$

which implies that $\mathtt{D}$ is isomorphic to $\mathtt{D} \to \mathtt{D}$.

A reflexive type can be used to interpret the untyped λ-calculus into the typed λ-calculus.

Untyped λ-calculus

We briefly describe the *untyped λ-calculus.* It is a theory whose terms are generated by the following grammar:

$$t ::= v \mid t_!\, t_2 \mid \lambda x.\, t\ .$$

In words, a variable is a term, an application $t\,t'$ is a term, for any terms t and t', and a λ-abstraction $\lambda x.\, t$ is a term, for any term t. Variable x is bound in $\lambda x.\, t$. A *context* is a list of distinct variables,

$$x_1, \ldots, x_n\ .$$

We say that a term t is valid in context Γ if the free variables of t are listed in Γ. The judgment that two terms u and t are equal is written as

$$\Gamma \mid u = t\ ,$$

where it is assumed that u and t are both valid in Γ. The context Γ is not really necessary but we include it because it is always good practice to list the free variables.

The rules of equality are as follows:

1. Equality is an equivalence relation:

$$\frac{}{\Gamma \mid t = t} \qquad \frac{\Gamma \mid t = u}{\Gamma \mid u = t} \qquad \frac{\Gamma \mid t = u \qquad \Gamma \mid u = v}{\Gamma \mid t = v}$$

2. The weakening rule:

$$\frac{\Gamma \mid u = t}{\Gamma, x \mid u = t}$$

3. Equations for application and λ-abstraction:

$$\frac{\Gamma \mid s = t \qquad \Gamma \mid u = v}{\Gamma \mid s\,u = t\,v} \qquad \frac{\Gamma, x \mid t = u}{\Gamma \mid \lambda x.\, t = \lambda x.\, u}$$

$$\frac{}{\Gamma \mid (\lambda x.\, t)u = t[u/x]} \tag*{(β-rule)}$$

$$\frac{}{\Gamma \mid \lambda x.\, (t\,x) = t} \ \text{ if } x \notin \mathsf{FV}(t) \tag*{(η-rule)}$$

The untyped λ-calculus can be translated into the theory of a reflexive type from Example 4.3.7. An untyped context Γ is translated to a typed context Γ^* by typing each variable in Γ with the reflexive type $\mathtt{D}$, i.e., a context $x_1, \ldots, x_k$ is translated to $x_1 : \mathtt{D}, \ldots, x_k : \mathtt{D}$. An untyped term t is translated to a typed term t^* as follows:

$$\begin{aligned} x^* &= x \qquad \text{if } x \text{ is a variable} , \\ (u\,t)^* &= (\mathtt{r}\,u^*)t^* , \\ (\lambda x.\,t)^* &= \mathtt{s}\,(\lambda x : \mathtt{D}\,.\,t^*) . \end{aligned}$$

For example, the term $\lambda x.\,(x\,x)$ translates to $\mathtt{s}\,(\lambda x : \mathtt{D}\,.\,((\mathtt{r}\,x)\,x))$. A judgment

$$\Gamma \mid u = t \tag{4.4}$$

is translated to the judgment

$$\Gamma^* \mid u^* = t^* : \mathtt{D} . \tag{4.5}$$

Exercise* 4.3.8. Prove that if equation (4.4) is provable then equation (4.5) is provable as well. Identify precisely at which point in your proof you need to use equations (4.2) and (4.3). Does provability of (4.5) imply provability of (4.4)?

Higher-order logic

This example presumes familiarity with the results of Chapter **??**, or at least with the basic categorical approach to first-order logic as presented in [**?**, **?**]. The approach to IHOL presented here is closely tied to *topos theory*, which is to be treated in greater depth in Chapter **??**.

To be added ...

4.4 Interpretation of λ-calculus in a CCC

We now consider semantic aspects of the λ-calculus and λ-theories. Suppose $\mathbb{T}$ is a λ-theory and $\mathcal{C}$ is a cartesian closed category. An *interpretation* $[\![-]\!]$ of $\mathbb{T}$ in $\mathcal{C}$ is given by the following data:

- For every basic type $\mathtt{B}$ in $\mathbb{T}$ an object $[\![\mathtt{B}]\!] \in \mathcal{C}$. The interpretation is extended to all types by

$$[\![\mathtt{1}]\!] = \mathbf{1} , \qquad [\![A \times B]\!] = [\![A]\!] \times [\![B]\!] , \qquad [\![A \to B]\!] = [\![B]\!]^{[\![A]\!]} .$$

- For every basic constant $\mathtt{c}$ of type C, a morphism $[\![\mathtt{c}]\!] : \mathbf{1} \to [\![\mathtt{C}]\!]$.

The interpretation is extended to all terms in context as follows.

- A context $\Gamma = x_1 : A_1, \cdots, x_n : A_n$ is interpreted as the object
$$\llbracket A_1 \rrbracket \times \cdots \times \llbracket A_n \rrbracket \,,$$
and the empty context is interpreted as the terminal object,
$$\llbracket \cdot \rrbracket = \mathsf{1}\,.$$
- A typing judgment
$$\Gamma \mid t : A$$
will be interpreted as a morphism
$$\llbracket \Gamma \mid t : A \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket \,.$$

The interpretation is defined inductively by the following rules:

- The i-th variable is interpreted as the i-th projection,
$$\llbracket x_0 : A_0, \ldots, x_n : A_n \mid x_i : A_i \rrbracket = \pi_i : \llbracket \Gamma \rrbracket \to \llbracket A_i \rrbracket \,.$$
- A basic constant $\mathtt{c} : \mathtt{C}$ in context Γ is interpreted as the composition
$$\begin{tikzcd} \llbracket \Gamma \rrbracket \arrow[r, "!_{\llbracket \Gamma \rrbracket}"] & \mathsf{1} \arrow[r, "\llbracket \mathtt{c} \rrbracket"] & \llbracket A \rrbracket \end{tikzcd}$$
- The interpretation of projections and pairs is
$$\begin{aligned} \llbracket \Gamma \mid \langle t, u \rangle : A \times B \rrbracket &= \langle \llbracket \Gamma \mid t : A \rrbracket, \llbracket \Gamma \mid u : B \rrbracket \rangle : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket \times \llbracket B \rrbracket \\ \llbracket \Gamma \mid \mathtt{fst}\, t : A \rrbracket &= \pi_0 \circ \llbracket \Gamma \mid t : A \times B \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket \\ \llbracket \Gamma \mid \mathtt{snd}\, t : A \rrbracket &= \pi_1 \circ \llbracket \Gamma \mid t : A \times B \rrbracket : \llbracket \Gamma \rrbracket \to \llbracket B \rrbracket \,. \end{aligned}$$
- The interpretation of application and λ-abstraction is
$$\begin{aligned} \llbracket \Gamma \mid t\, u : B \rrbracket &= \epsilon \circ \langle \llbracket \Gamma \mid t : A \to B \rrbracket, \llbracket \Gamma \mid u : A \rrbracket \rangle : \llbracket \Gamma \rrbracket \to \llbracket B \rrbracket \\ \llbracket \Gamma \mid \lambda x : A \,.\, t : A \to B \rrbracket &= (\llbracket \Gamma, x : A \mid t : B \rrbracket)^{\sim} : \llbracket \Gamma \rrbracket \to \llbracket B \rrbracket^{\llbracket A \rrbracket} \end{aligned}$$
where $\epsilon : \llbracket A \to B \rrbracket \times \llbracket A \rrbracket \to \llbracket B \rrbracket$ is the evaluation morphism for $\llbracket B \rrbracket^{\llbracket A \rrbracket}$ and $(\llbracket \Gamma, x : A \mid t : B \rrbracket)^{\sim}$ is the transpose of the morphism
$$\llbracket \Gamma, x : A \mid t : B \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \to \llbracket B \rrbracket \,.$$

Definition 4.4.1. An interpretation of a λ-theory $\mathbb{T}$ is a *model* of $\mathbb{T}$ if it *satisfies* all the axioms of $\mathbb{T}$, in the sense that for every axiom $\Gamma \mid u = v : A$ of $\mathbb{T}$, the interpretations of u and v coincide as arrows in $\mathcal{C}$,
$$\llbracket \Gamma \mid u : A \rrbracket = \llbracket \Gamma \mid v : A \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket A \rrbracket.$$

It follows that all equations that are provable in $\mathbb{T}$ are also satisfied in any model, by the following basic fact.

Proposition 4.4.2 (Soundness)**.** *If $\mathbb{T}$ is a λ-theory and $[\![-]\!]$ a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every equation in context $\Gamma \mid s = t : C$ that is provable from the axioms of $\mathbb{T}$, we have*

$$[\![\Gamma \mid s : C]\!] = [\![\Gamma \mid t : C]\!] : [\![\Gamma]\!] \longrightarrow [\![C]\!]\,.$$

Briefly, for all $\mathbb{T}$-models $[\![-]\!]$,

$$\mathbb{T} \vdash (\Gamma \mid s = t : C) \quad \textit{implies} \quad [\![-]\!] \models (\Gamma \mid s = t : C)\,.$$

The proof is a straightforward induction, first on the typing judgements for the interpretation, and then on the equational rules for the equations. If we stop after the first step, we can consider just the following notion of *inhabitation*:

Remark 4.4.3 (Inhabitation)**.** There is another notion of provability for the λ-calculus, related to the Curry-Howard correspondence of section 4.1, relating it to propositional logic. If we regard types as "propositions" rather than generalized algebraic structures, and terms as "proofs" rather than operations in such structures, then it is more natural to ask whether there even *is* a term $a : A$ of some type, than whether two terms of the same type are equal $s = t : A$. Of course, this only makes sense when A is considered in the empty context $\cdot \vdash A$, rather than $\Gamma \vdash A$ for non-empty Γ (consider the case where $\Gamma = x : A, \dots$). We say that a type A is *inhabited* (by a closed term) when there is some $\vdash a : A$, and regard an inhabited type A as one that is *provable*. There is then a different notion of soundness related to this notion of provability.

Proposition 4.4.4 (Inhabitation soundness)**.** *If $\mathbb{T}$ is a λ-theory and $[\![-]\!]$ a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every type A that is inhabited in $\mathbb{T}$, there is a point $1 \to [\![A]\!]$ in $\mathcal{C}$. Thus for all $\mathbb{T}$-models $[\![-]\!]$,*

$$\vdash a : A \quad \textit{implies there is a point} \;\; 1 \to [\![A]\!]\,.$$

This follows immediately from the fact that $[\![\cdot]\!] = 1$ for the empty context; for then the interpretation of any $\vdash a : A$ is a point

$$[\![a]\!] : 1 \to [\![A]\!]\,.$$

Example 4.4.5.

1. A model of an algebraic theory $\mathbb{A}$, extended to a λ-theory $\mathbb{A}^\lambda$ as in Example 4.3.3, taken in a CCC $\mathcal{C}$, is just a model of the algebraic theory $\mathbb{A}$ in the underlying finite product category $|\mathcal{C}|_\times$ of $\mathcal{C}$. An important difference, however, is that in defining the *category of models*

$$\mathsf{Mod}_{\mathsf{FP}}(\mathbb{A}, |\mathcal{C}|_\times)$$

we can take all homomorphisms of models of $\mathbb{A}$ as arrows, while the arrows in the category

$$\mathsf{Mod}_\lambda(\mathbb{A}^\lambda, \mathcal{C})$$

of λ-models are best taken to be isomorphisms, for which one has an obvious way to deal with the contravariance of the function type $[\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}$ (this is discussed in more detail in the next section).

2. A model of the theory of a reflexive type, Example 4.3.7, in Set must be the one-element set $1 = \{\star\}$ (prove this!). Fortunately, the exponentials in categories of presheaves are *not* computed pointwise; otherwise it would follow that this theory has no non-trivial models at all! (And then, by Theorem 4.7.6, that the theory itself is degenerate, in the sense that all equations are provable.) That there are non-trivial models is an important fact in the semantics of programming languages and the subject called *domain theory.* A fundamental paper in which this is shown is [**?**].

3. A (positive) propositional theory $\mathbb{T}$ may be regarded as a λ-theory, and a model in a cartesian closed poset P is then the same thing as before: an interpretation of the atomic propositions $p_1, p_2, ...$ of $\mathbb{T}$ as elements $[\![p_1]\!], [\![p_2]\!], ... \in P$, such that the axioms $\phi_1, \phi_2, ...$ of $\mathbb{T}$ are all sent to $1 \in P$ by the extension of $[\![-]\!]$ to all formulas,

$$1 = [\![\phi_1]\!] = [\![\phi_2]\!] = \cdots \in P\,.$$

Exercise 4.4.6. How are models of a (not necessarily propositional) λ-theory $\mathbb{T}$ in Cartesian closed *posets* related to models in arbitrary Cartesian closed categories? (*Hint:* Consider the inclusion $\mathsf{CCPos} \hookrightarrow \mathsf{CCC}$. Does it have any adjoints?)

4.5 Functorial semantics of STT in CCCs

In Chapter **??** we saw how algebraic theories can be viewed as categories (with finite products), and their algebras, or models, as functors (preserving finite products), and we arranged this analysis of the traditional relationship between syntax and sematics into a framework that we called *functorial semantics.* In Chapter **??**, we did the same for propositional logic. As a common generalization of both, the same framework of functorial semantics can be applied to λ-theories and their models in CCCs. The first step is to build a *classifying category* $\mathcal{C}_\mathbb{T}$ from a λ-theory $\mathbb{T}$, which again is constructed from the theory itself as a syntactic category. This is done as follows:

Definition 4.5.1. For any λ-theory $\mathbb{T}$, the *syntactic category* $\mathcal{C}_\mathbb{T}$ is determined as follows.

- The objects of $\mathcal{C}_\mathbb{T}$ are the types of $\mathbb{T}$.
- Arrows $A \to B$ are terms in context

$$[x : A \mid t : B]\ ,$$

where two such terms $x : A \mid s : B$ and $x : A \mid t : B$ represent the same morphism when $\mathbb{T}$ proves $x : A \mid s = t : B$.

- Composition of the terms

  $$[x : A \mid t : B] : A \longrightarrow B \qquad \text{and} \qquad [y : B \mid u : C] : B \longrightarrow C$$

  is the term obtained by substituting t for y in u:

  $$[x : A \mid u[t/y] : C] : A \longrightarrow C \; .$$

- The identity morphism on A is the term $[x : A \mid x : A]$.

Proposition 4.5.2. *The syntactic category $\mathcal{C}_{\mathbb{T}}$ built from a λ-theory is cartesian closed.*

Proof. We omit the equivalence classes brackets $[x : A \mid t : B]$ and simply treat equivalent terms as equal.

- The terminal object is the unit type $\mathtt{1}$. For any type A the unique morphism $!_A : A \to \mathtt{1}$ is the term

  $$x : A \mid * : \mathtt{1} \; .$$

  This morphism is indeed unique, because we have the equation

  $$\Gamma \mid t = * : \mathtt{1}$$

  is an axiom for the terms of unit type $\mathtt{1}$.

- The product of objects A and B is the type $A \times B$. The first and the second projections are the terms

  $$c : A \times B \mid \mathtt{fst}\, c : A \; , \qquad\qquad c : A \times B \mid \mathtt{snd}\, c : B \; .$$

  Given morphisms

  $$z : C \mid a : A \; , \qquad\qquad z : C \mid b : B \; ,$$

  the term

  $$z : C \mid \langle a, b \rangle : A \times B$$

  represents the unique morphism satisfying

  $$z : C \mid \mathtt{fst}\, \langle a, b \rangle = a : A \; , \qquad\qquad z : C \mid \mathtt{snd}\, \langle a, b \rangle = b : B \; .$$

  Indeed, if $\mathtt{fst}\, t = a$ and $\mathtt{snd}\, t = b$ for some t, then

  $$t = \langle \mathtt{fst}\, t, \mathtt{snd}\, t \rangle = \langle a, b \rangle \; .$$

- The exponential of objects A and B is the type $A \to B$ with the evaluation morphism

  $$e : (A \to B) \times A \mid (\mathtt{fst}\, e)(\mathtt{snd}\, e) : B\ .$$

  The transpose of a morphism $w : C \times A \mid t : B$ is the term

  $$z : C \mid \lambda x : A\,.\,(t[\langle z, x\rangle/w]) : A \to B\ .$$

  Showing that this is the transpose of t amounts to showing, in context $w : C \times A$,

  $$(\lambda x : A\,.\,(t[\langle \mathtt{fst}\, w, x\rangle/w]))(\mathtt{snd}\, w) = t : B$$

  Indeed, we have:

  $$(\lambda x : A\,.\,(t[\langle \mathtt{fst}\, w, x\rangle/w]))(\mathtt{snd}\, w) = t[\langle \mathtt{fst}\, w, \mathtt{snd}\, w\rangle/w] = t[w/w] = t\ ,$$

  which is a valid chain of equations in λ-calculus. The transpose is unique, because any morphism $z : C \mid s : A \to B$ that satisfies

  $$(s[\mathtt{fst}\, w/z])(\mathtt{snd}\, w) = t$$

  is equal to $\lambda x : A\,.\,(t[\langle z, x\rangle/w])$, because then

  $$\begin{aligned} t[\langle z, x\rangle/w] = (s[\mathtt{fst}\, w/z])(\mathtt{snd}\, w)[\langle z, x\rangle/w] = \\ (s[\mathtt{fst}\,\langle z, x\rangle/z])(\mathtt{snd}\,\langle z, x\rangle) = (s[z/z])\, x = s\, x\ . \end{aligned}$$

  Therefore,

  $$\lambda x : A\,.\,(t[\langle z, x\rangle/w]) = \lambda x : A\,.\,(s\, x) = s\ ,$$

  as claimed.

□

Now as before, the syntactic category allows us to replace a model $[\![-]\!]$ in a CCC $\mathcal{C}$ with a functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$. More precisely, we have the following.

Lemma 4.5.3. *A model $[\![-]\!]$ of a λ-theory $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$ determines a cartesian closed functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ with*

$$M(\mathtt{B}) = [\![\mathtt{B}]\!], \quad M(\mathtt{c}) = [\![\mathtt{c}]\!] : 1 \to [\![C]\!] = M(C)\,, \tag{4.6}$$

for all basic types $\mathtt{B}$ and basic constants $\mathtt{c} : C$. Moreover, M is unique up to a unique isomorphism of CCC functors, in the sense that given another model N satisfying (4.6), there is a unique natural iso $M \cong N$, determined inductively by the comparison maps $M(1) \cong N(1)$,

$$M(A \times B)\ \cong\ MA \times MB\ \cong\ NA \times NB\ \cong\ N(A \times B)\,,$$

and similarly for $M(B^A)$.

Proof. Straightforward. □

We then also have the usual functorial semantics theorem:

Theorem 4.5.4. *For any λ-theory $\mathbb{T}$, the syntactic category $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models, in the sense that for any cartesian closed category $\mathcal{C}$ there is an equivalence of categories*

$$\mathsf{Mod}_\lambda\big(\mathbb{T},\mathcal{C}\big) \;\simeq\; \mathsf{CCC}\big(\mathcal{C}_{\mathbb{T}},\mathcal{C}\big)\,, \tag{4.7}$$

naturally in $\mathcal{C}$. The morphisms of $\mathbb{T}$-models on the left are the isomorphisms of the underlying structures, and on the right we take the natural isomorphisms of CCC functors.

Proof. The only thing remaining to show is that, given a model $\llbracket - \rrbracket$ in a CCC $\mathcal{C}$ and a CCC functor $f : \mathcal{C} \to \mathcal{D}$, there is an induced model $\llbracket - \rrbracket^f$ in $\mathcal{D}$, given by the interpretation $\llbracket A \rrbracket^f = f\llbracket A \rrbracket$. This is straightforward, just as for algebraic theories. □

Remark 4.5.5. As mentioned in Example 4.4.5(1) the categories involved in the equivalence (4.7) are *groupoids*, in which every arrow is iso. The reason we have defined them as such is that the contravariant argument A in the function type $A \to B$ prevents us from specifying a non-iso homomorphism of models $h : M \to N$ by the obvious recursion on the type structure.

In more detail, given $h_A : \llbracket A \rrbracket^M \to \llbracket A \rrbracket^N$ and $h_B : \llbracket B \rrbracket^M \to \llbracket B \rrbracket^N$, there is no obvious candidate for a map

$$h_{A\to B} : \llbracket A \to B \rrbracket^M \longrightarrow \llbracket A \to B \rrbracket^N,$$

when all we have are the following induced maps:

$$\begin{tikzcd}
\llbracket A \to B \rrbracket^M \arrow[r, "="] & (\llbracket B \rrbracket^M)^{\llbracket A \rrbracket^M} \arrow[r, "(h_B)^{\llbracket A \rrbracket^M}"] & (\llbracket B \rrbracket^N)^{\llbracket A \rrbracket^M} & \\
& (\llbracket B \rrbracket^M)^{\llbracket A \rrbracket^N} \arrow[u, "(\llbracket B \rrbracket^M)^{h_A}"] \arrow[r, "(h_B)^{\llbracket A \rrbracket^N}"'] & (\llbracket B \rrbracket^N)^{\llbracket A \rrbracket^N} \arrow[u, "(\llbracket B \rrbracket^N)^{h_A}"'] \arrow[r, "="'] & \llbracket A \to B \rrbracket^N
\end{tikzcd}$$

One solution is therefore to take *isos* $h_A : \llbracket A \rrbracket^M \cong \llbracket A \rrbracket^N$ and $h_B : \llbracket B \rrbracket^M \cong \llbracket B \rrbracket^N$ and then use the inverses $h_A^{-1} : \llbracket A \rrbracket^N \to \llbracket A \rrbracket^M$ in the contravariant positions, in order to get things to line up:

$$\begin{tikzcd}
\llbracket A \to B \rrbracket^M \arrow[r, "="] & (\llbracket B \rrbracket^M)^{\llbracket A \rrbracket^M} \arrow[r, "(h_B)^{\llbracket A \rrbracket^M}"] \arrow[d, "(\llbracket B \rrbracket^M)^{h_A^{-1}}"', "\sim"] & (\llbracket B \rrbracket^N)^{\llbracket A \rrbracket^M} \arrow[d, "(\llbracket B \rrbracket^N)^{h_A^{-1}}", "\sim"'] & \\
& (\llbracket B \rrbracket^M)^{\llbracket A \rrbracket^N} \arrow[r, "(h_B)^{\llbracket A \rrbracket^N}"'] & (\llbracket B \rrbracket^N)^{\llbracket A \rrbracket^N} \arrow[r, "="'] & \llbracket A \to B \rrbracket^N
\end{tikzcd}$$

This suffices to at get a *category* of models $\mathsf{Mod}_\lambda\big(\mathbb{T},\mathcal{C}\big)$, rather than just as set, which is enough structure to determine the equivalence (4.7). Note that for an algebraic theory $\mathbb{A}$, this category of λ-models in Set, say, $\mathsf{Mod}_\lambda(\mathbb{A}^\lambda)$ is still the (wide but non-full) subcategory of isomorphisms of conventional (algebraic) $\mathbb{A}$-models

$$\mathsf{Mod}_\lambda(\mathbb{A}^\lambda) \rightarrowtail \mathsf{Mod}(\mathbb{A})\,.$$

We shall consider other solutions to the problem of contravariance below.

We can now proceed just as we did in the case of algebraic theories and prove that the semantics of λ-theories in cartesian closed categories is *complete*, in virtue of the syntactic construction of the classifying category $\mathcal{C}_\mathbb{T}$. Specifically, a λ-theory $\mathbb{T}$ has a canonical interpretation $[-]$ in the syntactic category $\mathcal{C}_\mathbb{T}$, which interprets a basic type A as itself, and a basic constant c of type A as the morphism $[x : \mathtt{1} \mid c : A]$. The canonical interpretation is a model of $\mathbb{T}$, also known as the *syntactic model*, in virtue of the definition of the equivalence relation $[-]$ on terms. In fact, it is a *logically generic* model of $\mathbb{T}$, because by the construction of $\mathcal{C}_\mathbb{T}$, for any terms $\Gamma \mid u : A$ and $\Gamma \mid t : A$, we have

$$\begin{aligned} \mathbb{T} \vdash (\Gamma \mid u = t : A) \quad &\Longleftrightarrow \quad [\Gamma \mid u : A] = [\Gamma \mid t : A] \\ &\Longleftrightarrow \quad [-] \models \Gamma \mid u = t : A\,. \end{aligned}$$

For the record, we therefore have now shown:

Proposition 4.5.6. *For any λ-theory $\mathbb{T}$,*

$$\mathbb{T} \vdash (\Gamma \mid t = u : A) \quad \textit{if, and only if,} \quad [-] \models (\Gamma \mid t = u : A) \textit{ for the syntactic model } [-].$$

Of course, the syntactic model $[-]$ is the one associated under (4.7) to the identity functor $\mathcal{C}_\mathbb{T} \to \mathcal{C}_\mathbb{T}$, *i.e.* it is the *universal* one. It therefore satisfies an equation just in case the equation holds in *all* models, by the classifying property of $\mathcal{C}_\mathbb{T}$, and the preservation of satisfaction of equations by CCC functors (Proposition 4.4.2).

Corollary 4.5.7. *For any λ-theory $\mathbb{T}$,*

$$\mathbb{T} \vdash (\Gamma \mid t = u : A) \quad \textit{if, and only if,} \quad M \models (\Gamma \mid t = u : A) \textit{ for every CCC model } M.$$

Moreover, a closed type A is inhabited $\vdash a : A$ if, and only if, there is a point $1 \to [\![A]\!]^M$ in every model M.

4.6 The internal language of a CCC

In the case of algebraic theories, we were able to recover the syntactic category from the semantics by taking certain Set-valued functors on the category of models in Set. This then extended to a duality between the category of all algebraic theories and that of all "algebraic categories", which we defined as the categories of Set-valued models of some algebraic theory (and also characterized abstractly). In the (classical) propositional case, this syntax-semantics duality was seen to be exactly the classical Stone duality between the categories of Boolean algebras and of Stone topological spaces. That sort of duality theory seems to be more difficult to formulate for λ-theories, however, now that we have taken the category of models to be just a groupoid (but see Remark **??**). Nonetheless, there is still a correspondence between λ-theories and CCCs, which we get by organizing the former into a category, which is then equivalent to that of the latter. But note that this is analogous to the equivalence between algebraic theories, regarded syntactically, and regarded as finite product categories—rather than to the duality between syntax and semantics.

In order to define the equivalence in question, we first need a suitable notion of *morphism of theories*. A *translation* $\tau : \mathbb{S} \to \mathbb{T}$ of a λ-theory $\mathbb{S}$ into a λ-theory $\mathbb{T}$ is given by the following data:

1. For each basic type A in $\mathbb{S}$ a type τA in $\mathbb{T}$. The translation is then extended to all types by the rules
$$\tau \mathsf{1} = \mathsf{1}\ , \qquad \tau(A \times B) = \tau A \times \tau B\ , \qquad \tau(A \to B) = \tau A \to \tau B\ .$$
2. For each basic constant c of type A in $\mathbb{S}$ a term τc of type τA in $\mathbb{T}$. The translation of terms is then extended to all terms by the rules
$$\begin{aligned} \tau(\mathtt{fst}\, t) &= \mathtt{fst}\,(\tau t)\ , & \tau(\mathtt{snd}\, t) &= \mathtt{snd}\,(\tau t)\ , \\ \tau\langle t, u\rangle &= \langle \tau t, \tau u\rangle\ , & \tau(\lambda x : A\,.\, t) &= \lambda x : \tau A\,.\,\tau t\ , \\ \tau(t\, u) &= (\tau t)(\tau u)\ , & \tau x &= x \quad \text{(if x is a variable)}\ . \end{aligned}$$

A context $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ is translated by τ to the context
$$\tau\Gamma = x_1 : \tau A_1, \ldots, x_n : \tau A_n\ .$$

Furthermore, a translation is required to preserve the axioms of $\mathbb{S}$: if $\Gamma \mid t = u : A$ is an axiom of $\mathbb{S}$ then $\mathbb{T}$ proves $\tau\Gamma \mid \tau t = \tau u : \tau A$. It then follows that all equations proved by $\mathbb{S}$ are translated to valid equations in $\mathbb{T}$.

A moment's consideration shows that a translation $\tau : \mathbb{S} \to \mathbb{T}$ is the same thing as a model of $\mathbb{S}$ in $\mathcal{C}_{\mathbb{T}}$, despite being specified entirely syntactically. More precisely, λ-theories and translations between them clearly form a category: translations compose as functions, therefore composition is associative. The identity translation $\iota_{\mathbb{T}} : \mathbb{T} \to \mathbb{T}$ translates every type to itself and every constant to itself.

Definition 4.6.1. Let $\lambda\mathsf{Thr}$ be the category whose objects are λ-theories and morphisms are translations between them.

We now have an *isomorphism of sets*,
$$\mathsf{Hom}_{\lambda\mathsf{Thr}}(\mathbb{S}, \mathbb{T}) \cong \mathsf{Mod}_{\lambda}(\mathbb{S}, \mathcal{C}_{\mathbb{T}})\ , \tag{4.8}$$

which is natural in the theory $\mathbb{S}$, as can be seen by considering the canonical interpretation of $\mathbb{S}$ in $\mathcal{C}_{\mathbb{S}}$ induced by the identity translation $\iota_{\mathbb{S}} : \mathbb{S} \to \mathbb{S}$.

Let $\mathcal{C}$ be a small cartesian closed category. There is a λ-theory $\mathbb{L}(\mathcal{C})$ corresponding to $\mathcal{C}$, called the *internal language of* $\mathcal{C}$, and defined as follows:

1. For every object $A \in \mathcal{C}$ there is a basic type $\ulcorner A \urcorner$.

2. For every morphism $f : A \to B$ there is a basic constant $\ulcorner f \urcorner$ whose type is $\ulcorner A \urcorner \to \ulcorner B \urcorner$.

3. For every $A \in \mathcal{C}$ there is an axiom
$$x : \ulcorner A \urcorner \mid \ulcorner 1_A \urcorner x = x : \ulcorner A \urcorner \,.$$

4. For all morphisms $f : A \to B$, $g : B \to C$, and $h : A \to C$ such that $h = g \circ f$, there is an axiom
$$x : \ulcorner A \urcorner \mid \ulcorner h \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner C \urcorner \,.$$

5. There is a constant
$$\mathtt{T} : \mathtt{1} \to \ulcorner 1 \urcorner \,,$$
and for all $A, B \in \mathcal{C}$ there are constants
$$\mathtt{P}_{A,B} : \ulcorner A \urcorner \times \ulcorner B \urcorner \to \ulcorner A \times B \urcorner \,, \qquad \mathtt{E}_{A,B} : (\ulcorner A \urcorner \to \ulcorner B \urcorner) \to \ulcorner B^A \urcorner \,.$$
They satisfy the following axioms:
$$\begin{aligned}
u : \ulcorner 1 \urcorner &\mid \mathtt{T} * = u : \ulcorner 1 \urcorner \\
z : \ulcorner A \times B \urcorner &\mid \mathtt{P}_{A,B} \langle \ulcorner \pi_0 \urcorner z, \ulcorner \pi_1 \urcorner z \rangle = z : \ulcorner A \times B \urcorner \\
w : \ulcorner A \urcorner \times \ulcorner B \urcorner &\mid \langle \ulcorner \pi_0 \urcorner (\mathtt{P}_{A,B} w), \ulcorner \pi_1 \urcorner (\mathtt{P}_{A,B} w) \rangle = w : \ulcorner A \urcorner \times \ulcorner B \urcorner \\
f : \ulcorner B^A \urcorner &\mid \mathtt{E}_{A,B} (\lambda x : \ulcorner A \urcorner \,.\, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle f, x \rangle))) = f : \ulcorner B^A \urcorner \\
f : \ulcorner A \urcorner \to \ulcorner B \urcorner &\mid \lambda x : \ulcorner A \urcorner \,.\, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle (\mathtt{E}_{A,B} f), x \rangle)) = f : \ulcorner A \urcorner \to \ulcorner B \urcorner
\end{aligned}$$

The purpose of the constants $\mathtt{T}$, $\mathtt{P}_{A,B}$, $\mathtt{E}_{A,B}$, and the axioms for them is to ensure the isomorphisms $\ulcorner 1 \urcorner \cong \mathtt{1}$, $\ulcorner A \times B \urcorner \cong \ulcorner A \urcorner \times \ulcorner B \urcorner$, and $\ulcorner B^A \urcorner \cong \ulcorner A \urcorner \to \ulcorner B \urcorner$. Types A and B are said to be *isomorphic* if there are terms
$$x : A \mid t : B \,, \qquad y : B \mid u : A \,,$$
such that $\mathbb{S}$ proves
$$x : A \mid u[t/y] = x : A \,, \qquad y : B \mid t[u/x] = y : B \,.$$
Furthermore, an *equivalence of theories* $\mathbb{S}$ and $\mathbb{T}$ is a pair of translations
$$\begin{tikzcd}
\mathbb{S} \arrow[r, bend left, "\tau"] & \mathbb{T} \arrow[l, bend left, "\sigma"]
\end{tikzcd}$$

such that, for any type A in $\mathbb{S}$ and any type B in $\mathbb{T}$,

$$\sigma(\tau A) \cong A\,, \qquad\qquad \tau(\sigma B) \cong B\,.$$

The assignment $\mathcal{C} \mapsto \mathbb{L}(\mathcal{C})$ extends to a functor

$$\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}\,,$$

where CCC is the category of small cartesian closed categories and functors between them that preserve finite products and exponentials. Such functors are also called *cartesian closed functors* or *ccc functors*. If $F : \mathcal{C} \to \mathcal{D}$ is a cartesian closed functor then $\mathbb{L}(F) : \mathbb{L}(\mathcal{C}) \to \mathbb{L}(\mathcal{D})$ is the translation given by:

1. A basic type $\ulcorner A \urcorner$ is translated to $\ulcorner FA \urcorner$.

2. A basic constant $\ulcorner f \urcorner$ is translated to $\ulcorner Ff \urcorner$.

3. The basic constants $\mathtt{T}$, $\mathtt{P}_{A,B}$ and $\mathtt{E}_{A,B}$ are translated to $\mathtt{T}$, $\mathtt{P}_{FA,BA}$ and $\mathtt{E}_{FA,FB}$, respectively.

We now have a functor $\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}$. How about the other direction? We already have the construction of syntactic category which maps a λ-theory $\mathbb{S}$ to a small cartesian closed category $\mathcal{C}_{\mathbb{S}}$. This extends to a functor

$$\mathcal{C} : \lambda\mathsf{Thr} \to \mathsf{CCC}\,,$$

because a translation $\tau : \mathbb{S} \to \mathbb{T}$ induces a functor $\mathcal{C}_\tau : \mathcal{C}_{\mathbb{S}} \to \mathcal{C}_{\mathbb{T}}$ in an obvious way: a basic type $A \in \mathcal{C}_{\mathbb{S}}$ is mapped to the object $\tau A \in \mathcal{C}_{\mathbb{T}}$, and a basic constant $x : \mathtt{1} \mid c : A$ is mapped to the morphism $x : \mathtt{1} \mid \tau c : A$. The rest of $\mathcal{C}_\tau$ is defined inductively on the structure of types and terms.

Theorem 4.6.2. *The functors $\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}$ and $\mathcal{C} : \lambda\mathsf{Thr} \to \mathsf{CCC}$ constitute an equivalence of categories "up to equivalence" (a* biequivalence *of* 2-categories*). This means that for any $\mathcal{C} \in \mathsf{CCC}$ there is an equivalence of categories*

$$\mathcal{C} \simeq \mathcal{C}_{\mathbb{L}(\mathcal{C})}\,,$$

and for any $\mathbb{S} \in \lambda\mathsf{Thr}$ there is an equivalence of theories

$$\mathbb{S} \simeq \mathbb{L}(\mathcal{C}_{\mathbb{S}})\,.$$

Proof. For a small cartesian closed category $\mathcal{C}$, consider the functor $\eta_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}_{\mathbb{L}(\mathcal{C})}$, defined for an object $A \in \mathcal{C}$ and $f : A \to B$ in $\mathcal{C}$ by

$$\eta_{\mathcal{C}} A = \ulcorner A \urcorner\,, \qquad\qquad \eta_{\mathcal{C}} f = (x : \ulcorner A \urcorner \mid \ulcorner f \urcorner x : \ulcorner B \urcorner)\,.$$

To see that $\eta_{\mathcal{C}}$ is a functor, observe that $\mathbb{L}(\mathcal{C})$ proves, for all $A \in \mathcal{C}$,

$$x : \ulcorner A \urcorner \mid \ulcorner \mathtt{1}_A \urcorner x = x : \ulcorner A \urcorner$$

and for all $f : A \to B$ and $g : B \to C$,

$$x : \ulcorner A \urcorner \mid \ulcorner g \circ f \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner C \urcorner \,.$$

To see that $\eta_{\mathcal{C}}$ is an equivalence of categories, it suffices to show that for every object $X \in \mathcal{C}_{\mathbb{L}(\mathcal{C})}$ there exists an object $\theta_{\mathcal{C}} X \in \mathcal{C}$ such that $\eta_{\mathcal{C}}(\theta_{\mathcal{C}} X) \cong X$. The choice map $\theta_{\mathcal{C}}$ is defined inductively by

$$\theta_{\mathcal{C}} \mathtt{1} = \mathtt{1} \,, \qquad \theta_{\mathcal{C}} \ulcorner A \urcorner = A \,,$$
$$\theta_{\mathcal{C}}(Y \times Z) = \theta_{\mathcal{C}} X \times \theta_{\mathcal{C}} Y \,, \qquad \theta_{\mathcal{C}}(Y \to Z) = (\theta_{\mathcal{C}} Z)^{\theta_{\mathcal{C}} Y} \,.$$

We skip the verification that $\eta_{\mathcal{C}}(\theta_{\mathcal{C}} X) \cong X$. In fact, $\theta_{\mathcal{C}}$ can be extended to a functor $\theta_{\mathcal{C}} : \mathcal{C}_{\mathbb{L}(\mathcal{C})} \to \mathcal{C}$ so that $\theta_{\mathcal{C}} \circ \eta_{\mathcal{C}} \cong 1_{\mathcal{C}}$ and $\eta_{\mathcal{C}} \circ \theta_{\mathcal{C}} \cong 1_{\mathcal{C}_{\mathbb{L}(\mathcal{C})}}$.

Given a λ-theory $\mathbb{S}$, we define a translation $\tau_{\mathbb{S}} : \mathbb{S} \to \mathbb{L}(\mathcal{C}_{\mathbb{S}})$. For a basic type A let

$$\tau_{\mathbb{S}} A = \ulcorner A \urcorner \,.$$

The translation $\tau_{\mathbb{S}} c$ of a basic constant c of type A is

$$\tau_{\mathbb{S}} c = \ulcorner x : \mathtt{1} \mid c : \tau_{\mathbb{S}} A \urcorner \,.$$

In the other direction we define a translaton $\sigma_{\mathbb{S}} : \mathbb{L}(\mathcal{C}_{\mathbb{S}}) \to \mathbb{S}$ as follows. If $\ulcorner A \urcorner$ is a basic type in $\mathbb{L}(\mathcal{C}_{\mathbb{S}})$ then

$$\sigma_{\mathbb{S}} \ulcorner A \urcorner = A \,,$$

and if $\ulcorner x : A \mid t : B \urcorner$ is a basic constant of type $\ulcorner A \urcorner \to \ulcorner B \urcorner$ then

$$\sigma_{\mathbb{S}} \ulcorner x : A \mid t : B \urcorner = \lambda x : A \,.\, t \,.$$

The basic constants $\mathtt{T}$, $\mathtt{P}_{A,B}$ and $\mathtt{E}_{A,B}$ are translated by $\sigma_{\mathbb{S}}$ into

$$\sigma_{\mathbb{S}}\, \mathtt{T} = \lambda x : \mathtt{1} \,.\, x \,,$$
$$\sigma_{\mathbb{S}}\, \mathtt{P}_{A,B} = \lambda p : A \times B \,.\, p \,,$$
$$\sigma_{\mathbb{S}}\, \mathtt{E}_{A,B} = \lambda f : A \to B \,.\, f \,.$$

If A is a type in $\mathbb{S}$ then $\sigma_{\mathbb{S}}(\tau_{\mathbb{S}} A) = A$. For the other direction, we would like to show, for any type X in $\mathbb{L}(\mathcal{C}_{\mathbb{S}})$, that $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}} X) \cong X$. We prove this by induction on the structure of type X:

1. If $X = \mathtt{1}$ then $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}} \mathtt{1}) = \mathtt{1}$.

2. If $X = \ulcorner A \urcorner$ is a basic type then A is a type in $\mathbb{S}$. We proceed by induction on the structure of A:

   (a) If $A = \mathtt{1}$ then $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}} \ulcorner \mathtt{1} \urcorner) = \mathtt{1}$. The types $\mathtt{1}$ and $\ulcorner \mathtt{1} \urcorner$ are isomorphic via the constant $\mathtt{T} : \mathtt{1} \to \ulcorner \mathtt{1} \urcorner$.

(b) If A is a basic type then $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}\ulcorner A\urcorner) = \ulcorner A\urcorner$.

(c) If $A = B \times C$ then $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}\ulcorner B \times C\urcorner) = \ulcorner B\urcorner \times \ulcorner C\urcorner$. But we know $\ulcorner B\urcorner \times \ulcorner C\urcorner \cong \ulcorner B \times C\urcorner$ via the constant $\mathsf{P}_{A,B}$.

(d) The case $A = B \to C$ is similar.

3. If $X = Y \times Z$ then $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}(Y \times Z)) = \tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Y) \times \tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Z)$. By induction hypothesis, $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Y) \cong Y$ and $\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Z) \cong Z$, from which we easily obtain

$$\tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Y) \times \tau_{\mathbb{S}}(\sigma_{\mathbb{S}}Z) \cong Y \times Z\ .$$

4. The case $X = Y \to Z$ is similar.

□

Composing the isomorphism 4.8 with the equivalence 4.7 we can formulate the foregoing Theorem 4.6.2 as an adjoint equivalence.

Corollary 4.6.3. *There is a biequivalence between the categories* $\lambda\mathsf{Thr}$ *of* λ*-theories and translations between them (and isos thereof), and the category* CCC *of cartesian closed categories and CCC functors (and natural isos),*

$$\begin{aligned} \mathsf{Hom}_{\lambda\mathsf{Thr}}\big(\mathbb{T}, \mathbb{L}\mathcal{C}\big) &\cong \mathsf{Mod}_{\lambda}\big(\mathbb{T}, \mathcal{C}\big)\,, \\ &\simeq\ \mathsf{Hom}_{\mathsf{CCC}}\big(\mathcal{C}_{\mathbb{T}}\,, \mathcal{C}\big)\,. \end{aligned}$$

This is mediated by an adjunction,

$$\begin{tikzcd} \mathsf{CCC} \arrow[r, bend left, "\mathbb{L}"] & \lambda\mathsf{Thr} \arrow[l, bend left, "\mathcal{C}"] \end{tikzcd}$$

with $\mathcal{C} \dashv \mathbb{L}$*, between the* syntactic category *functor* $\mathcal{C}$ *and the* internal language *functor* $\mathbb{L}$*.*

Exercise 4.6.4. In the proof of Theorem 4.6.2 we defined, for each $\mathcal{C} \in \mathsf{CCC}$, a functor $\eta_C : \mathcal{C} \to \mathcal{C}_{\mathbb{L}(\mathcal{C})}$. Verify that this determines a natural transformation $\eta : 1_{\mathsf{CCC}} \Longrightarrow \mathcal{C} \circ \mathbb{L}$ which is an equivalence of categories. What about the translation $\epsilon_{\mathbb{T}} : \mathbb{T} \to \mathbb{L}(\mathcal{C}_{\mathbb{T}})$—is that an isomorphism?

See the book [**?**] for another approach to the biequivalence of Corollary 4.6.3, which turns it into an equivalence of categories by fixing the CCC structure and requiring it to be preserved *strictly*.

4.7 Embedding and completeness theorems

We have considered the λ-calculus as a common generalization of both propositional logic, modelled by poset CCCs such as Boolean and Heyting algebras, and equational logic, modelled by finite product categories. Accordingly, there are then two different notions of "provability", as discussied in Remark 4.4.3; namely, the derivability of a closed term $\vdash a : A$, and the derivability of an equation between two (not necessarily closed) terms of the same type $\Gamma \vdash s = t : A$. With respect to the semantics, there are then two different corresponding notions of soundness and completeness: for "inhabitation" of types, and for equality of terms. We consider special cases of these notions in more detail below.

Conservativity

With regard to the former notion, inhabitation, one can also consider the question of how it compares with simple provability in *propositional logic*: e.g. a positive propositional formula ϕ in the variables $p_1, p_2, ..., p_n$ obviously determines a type Φ in the corresponding λ-theory $\mathbb{T}(X_1, X_2, ..., X_n)$ over n basic type symbols. What is the relationship between provability in positive propositional logic, $\mathsf{PPL} \vdash \phi$, and inhabitation in the associated λ-theory, $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$? Let us call this the question of *conservativity* of λ-calculus over PPL. According to the basic idea of the Curry-Howard correspondence from Section 4.1, the λ-calculus is essentially the "proof theory of PPL". So one should expect that starting from an inhabited type Φ, a derivation of a term $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$ should result in a corresponding proof of ϕ in PPL just by "rubbing out the proof terms". Conversely, given a provable formula $\vdash \phi$, one should be able to annotate a proof of it in PPL to obtain a derivation of a term $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$ in the λ-calculus (although perhaps not the same term that one started with, if the proof was obtained from rubbing out a term).

We can make this idea precise semantically as follows. Write $|\mathcal{C}|$ for the poset reflection of a category $\mathcal{C}$, that is, the left adjoint to the inclusion $i : \mathsf{Pos} \hookrightarrow \mathsf{Cat}$, and let $\eta : \mathcal{C} \to |\mathcal{C}|$ be the unit of the adjunction.

Lemma 4.7.1. *If $\mathcal{C}$ is cartesian closed, then so is $|\mathcal{C}|$, and $\eta : \mathcal{C} \to |\mathcal{C}|$ preserves the CCC structure.*

Proof. Exercise! □

Exercise 4.7.2. Prove Lemma 4.7.1.

Corollary 4.7.3. *The syntactic category $\mathsf{PPC}(p_1, p_2, ..., p_n)$ of the positive propositional calculus on n propositional variables is the poset reflection of the syntactic category $\mathcal{C}_{\mathbb{T}(X_1, X_2, ..., X_n)}$ of the λ-theory $\mathbb{T}(X_1, X_2, ..., X_n)$,*

$$|\mathcal{C}_{\mathbb{T}(X_1, X_2, ..., X_n)}| \cong \mathsf{PPC}(p_1, p_2, ..., p_n)\,.$$

Proof. We already know that $\mathcal{C}_{\mathbb{T}(X_1,X_2,\ldots,X_n)}$ is the free cartesian closed category on n generating objects, and that $\mathsf{PPC}(p_1, p_2, \ldots, p_n)$ is the free cartesian closed poset on n generating elements. From the universal property of $\mathcal{C}_{\mathbb{T}(X_1,X_2,\ldots,X_n)}$, we get a CCC map

$$\mathcal{C}_{\mathbb{T}(X_1,X_2,\ldots,X_n)} \longrightarrow \mathsf{PPC}(p_1, p_2, \ldots, p_n)$$

taking generators to generators, and it extends along the quotient map to $|\mathcal{C}_{\mathbb{T}(X_1,X_2,\ldots,X_n)}|$ by the universal property of the poset reflection. Thus it suffices to show that the quotient map preserves, and indeed creates, the CCC structure on $|\mathcal{C}_{\mathbb{T}(X_1,X_2,\ldots,X_n)}|$, which follows from the Lemma 4.7.1. □

Remark 4.7.4. Corollary 4.7.3 can be extended to other systems of type theory and logic, with further operations such as CCCs with sums $0, A+B$ ("bicartesian closed categories"), and the full intuitionistic propositional calculus IPC with the logical operations $\bot$ and $p \vee q$. We leave this as a topic for the interested student.

Completeness

As was the case for equational theories and propositional logic, the fact that there is a generic model (Proposition 4.5.6) allows the general completeness theorem stated in Corollary 4.5.7 to be specialized to various classes of special models, via embedding (or "representation") theorems, this time for CCCs, rather than for finite product categories or Boolean/Heyting algebras. We shall consider three such cases: "variable" models, Kripke models, and topological models. In each case, an "embedding theorem" of the form:

Every CCC embeds into one of the special form $\mathcal{X}$.

gives rise to a completeness theorem of the form:

For all λ-theories $\mathbb{T}$, if $1 \to [\![A]\!]^M$ in all $\mathbb{T}$-models M in all $\mathcal{X}$, then $\mathbb{T} \vdash a : A$,
and if $[\![a]\!]^M = [\![b]\!]^M : 1 \to [\![A]\!]$ in all $\mathbb{T}$-models M in all $\mathcal{X}$, then $\mathbb{T} \vdash a = b : A$.

This of course follows the same pattern that we saw for the simpler "proof relevant" case of equational (*i.e.* finite product) theories, and the even simpler "proof irrelevant" case of propositional logic, but now the proofs of some of the embedding theorems for CCCs require more sophisticated methods.

Variable models

By a *variable model* of the λ-calculus we mean one in a CCC of the form $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, i.e. presheaves on a (small) category $\mathbb{C}$. We regard such a model as "varying over $\mathbb{C}$", just as we saw earlier that a presheaf of groups on *e.g.* the simplex category Δ may be seen both as a simplicial group—a simplicial object in the category of groups—and as a group in the category $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ of simplicial sets. The basic embedding theorem that we use in specializing Proposition 4.5.6 to such variable models is the following, which is one of the fundamental facts of categorical semantics.

Lemma 4.7.5. *For any small cartesian closed category* $\mathbb{C}$*, the Yoneda embedding*

$$\mathsf{y} : \mathbb{C} \hookrightarrow \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$$

preserves the cartesian closed structure.

This is of course the "categorified" analogue of Lemma 2.8.5, which we used for the Kripke completeness of the positive propositional calculus PPC.

Proof. We can just evaluate $\mathsf{y}A(X) = \mathbb{C}(X, A)$. It is clear that $\mathsf{y}1(X) = \mathbb{C}(X, 1) \cong 1$ naturally in X, and that $\mathsf{y}(A \times B)(X) = \mathbb{C}(X, A \times B) \cong \mathbb{C}(X, A) \times \mathbb{C}(X, B) \cong (\mathsf{y}A \times \mathsf{y}B)(X)$ for all A, B, X, naturally in all three arguments. For $B^A \in \mathbb{C}$, we then have

$$\mathsf{y}(B^A)(X) = \mathbb{C}(X, B^A) \cong \mathbb{C}(X \times A, B) \cong \widehat{\mathbb{C}}(\mathsf{y}(X \times A), \mathsf{y}B) \cong \widehat{\mathbb{C}}(\mathsf{y}X \times \mathsf{y}A, \mathsf{y}B),$$

since y is full and faithful and, as we just showed, preserves $\times$. But now recall that the exponential Q^P of presheaves P, Q is defined at X by the specification

$$Q^P(X) \;=\; \widehat{\mathbb{C}}(\mathsf{y}X \times P, Q)\,.$$

So, continuing where we left off, $\widehat{\mathbb{C}}(\mathsf{y}X \times \mathsf{y}A, \mathsf{y}B) = \mathsf{y}B^{\mathsf{y}A}(X)$, and we're done. □

For an early version of the following theorem (and much more), see the nice paper [Sco80] by Dana Scott.

Theorem 4.7.6. *For any* λ*-theory* $\mathbb{T}$*, we have the following:*

(i) A type A *is inhabited,*

$$\mathbb{T} \vdash a : A$$

if, and only if, for every a small category $\mathbb{C}$*, in every model* $[\![-]\!]$ *in presheaves* $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ *on* $\mathbb{C}$*, there is a point*

$$1 \to [\![A]\!]\,.$$

(ii) For any terms $\Gamma \mid s, t : A$*,*

$$\mathbb{T} \vdash (\Gamma \mid s = t : A)$$

if, and only if,

$$[\![\Gamma \vdash s : A]\!] = [\![\Gamma \vdash t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$$

for every such presheaf model.

Proof. We simply specialize the general completeness statement of Corollary 4.5.7 to CCCs of the form $\widehat{\mathbb{C}}$ using Lemma 4.7.5, together with the fact that the Yoneda embedding is full (and therefore reflects inhabitation) and faithful (and therefore reflects satisfaction of equations). □

4.8 Kripke models

By a Kripke model of (a theory $\mathbb{T}$ in) the λ-calculus, we mean a model $[\![-]\!]$ in the sense of Definition 4.4.1 in a presheaf CCC of the form Set^K for a poset K, *i.e.* a variable model in the sense of the previous section, where the domain of variation is just a *poset*, rather than a proper category. As with Kirpke models of propositional logic, we can regard such a model as varying through (branching) time, over a causally ordered state space, or some other partially-ordered parameter space. By Theorem 4.5.4, such a model $(K, [\![-]\!])$ is essentially the same thing as a CCC functor $M : \mathcal{C}_{\mathbb{T}} \to \mathsf{Set}^K$, taking values in "variable sets". Regarding the λ-calculus as the proof theory of the propositional calculus via the Curry-Howard correspondence (Section 4.1), it is perhaps not surprising that it should be (inhabitation) *complete* with respect to such Kripke models, in light of Theorem 2.8.4. Completeness with respect to *equations between terms* is another matter, though; while true, the proof is far from a simple generalization of other known results. It can perhaps be seen as a verification that $\beta\eta$-equivalence is the "right" notion of equality for proofs.

Before considering such questions, however, let us first spell out explicitly what such a Kripke model looks like for the simple example of a theory $\mathbb{T}$ of an object with a commutative, binary operation,

$$\mathbb{T} \;=\; \big(\mathtt{B},\; m : \mathtt{B} \times \mathtt{B} \to \mathtt{B},\; x * y = y * x\big)\,.$$

There is one basic type symbol $\mathtt{B}$, a binary operation symbol $* : \mathtt{B} \times \mathtt{B} \to \mathtt{B}$, and a single equation $x, y : \mathtt{B} \,|\, x * y = y * x : \mathtt{B}$. Let K be a poset with ordering relation $j \leq k$ for $j, k \in K$.

A *Kripke model M of $\mathbb{T}$ over K* consists, first, of a family of sets $(M_k)_{k \in K}$, equipped with functions

$$m_{j,k} : M_j \to M_k \qquad (\text{for all } j \leq k \in K)\,,$$

satisfying the conditions:

$$m_{k,k} = 1_{M_k}, \qquad m_{j,k} \circ m_{i,j} = m_{i,k} \qquad (\text{for all } j \leq k \in K)\,.$$

This is of course exactly a functor $M : K \to \mathsf{Set}$, as the interpretation $M = [\![\mathtt{B}]\!]$ of the basic type symbol $\mathtt{B}$. Next, we need functions

$$s_k : M_k \times M_k \to M_k \qquad (\text{for all } k \in K)$$

satisfying

$$s_k\big(m_{j,k}(x), m_{j,k}(y)\big) \;=\; m_{j,k}\big(s_j(x,y)\big) \qquad (\text{for all } j \leq k \in K \text{ and } x, y \in M_j)\,.$$

This is just a natural transformation $s : M \times M \to M$, as the interpretation $s = [\![*]\!]$ of the operation symbol $* : \mathtt{B} \times \mathtt{B} \to \mathtt{B}$. Finally, the interpretation $(M, s) = [\![\mathtt{B}, *]\!]$ should satisfy the equation $x, y : \mathtt{B} \,|\, x * y = y * x : \mathtt{B}$, meaning that

$$s_k(x, y) = s_k(y, x) \qquad (\text{for all } k \in K)\,,$$

since two natural transformations are equal just if all of their components are equal. Thus a Kripke model of this theory $\mathbb{T}$ is just a model of the underlying algebraic theory in the functor category Set^K—which of course is the same thing as a functor from K to the category of $\mathbb{T}$-models in Set.

A theory involving an operation of "higher type", such as the section $s : (D \to D) \to D$ in (the theory of) a reflexive type (Example 4.3.7) is no more "non-standard". Let $D = [\![\mathtt{D}]\!]$ be the interpretation of the basic type $\mathtt{D}$, so that $[\![\mathtt{D} \to \mathtt{D}]\!] = D^D : K \to \mathsf{Set}$ is an exponential presheaf. At each $k \in K$, we then have,

$$(D^D)_k = \mathsf{Set}^K\big(D \times K(k,-), D\big)\,,$$

which is trivial except on the upset $\uparrow k$, where it consists of natural transformations

$$\mathsf{Set}^{\uparrow k}\big(D{\uparrow}k, D{\uparrow}k\big)\,,$$

where $D{\uparrow}k : {\uparrow}k \to \mathsf{Set}$ is just D restricted to the upset ${\uparrow}k \subseteq K$, *i.e.* the composite

$${\uparrow}k \hookrightarrow K \xrightarrow{D} \mathsf{Set}\,.$$

Given any such natural transformation $\vartheta : D{\uparrow}k \longrightarrow D{\uparrow}k$, and any $k \leq j$, the action of the functor,

$$(D^D)_k \to (D^D)_j$$

on ϑ is simply to restrict it further to ${\uparrow}j \subseteq {\uparrow}k$, thus taking ϑ to

$$\vartheta{\uparrow}j : D{\uparrow}j \longrightarrow D{\uparrow}j\,,$$

which is just the same function as ϑ, with the new domain of definition ${\uparrow}j \subseteq {\uparrow}k$.

The section $s : (D \to D) \to D$ therefore takes, at each $k \in K$, such a $\vartheta : D{\uparrow}k \longrightarrow D{\uparrow}k$ to an element $s_k(\vartheta) \in D_k$, respecting the restrictions ${\uparrow}j \subseteq {\uparrow}k$ in the sense that

$$d_{k,j}s_k(\vartheta) = s_j(\vartheta{\uparrow}j) \in D_j\,,$$

where $d_{k,j} : D_k \to D_j$ is the action of the functor $D : K \to \mathsf{Set}$.

In this way, the presheaf exponential $D^D : K \to \mathsf{Set}$ is entirely determined by the "base-case" $D : K \to \mathsf{Set}$, and is still a "full function space" at each $k \in K$, but the functorial action in k requires it not to be just $D_k^{D_k}$ (which for a reflexive type would then be trivial at all $k \in K$), but rather, to take the entire segment ${\uparrow}k$ into account—much in the way that $k \Vdash \varphi \Rightarrow \psi$ was determined for Kripke models of the intuitionistic propositional calculus IPC by considering all $j \geq k$. (Indeed, one can explicitly formulate the Kripke semantics for simple type theory in the usual Kripke-forcing style $k \Vdash a : A$, *cf.* [AGH21].)

The proof of the following theorem uses a deep result from topos theory (due to Joyal-Tierney [**?**]) that is, unfortunately, beyond the scope of this book. It implies that, for every small CCC $\mathcal{C}$ there is a poset K and a full and faithful CCC functor $\mathcal{C} \hookrightarrow \mathsf{Set}^K$.

Theorem 4.8.1 (Kripke completeness for λ-calculus)**.** *For any λ-theory $\mathbb{T}$:*

(i) A type A is inhabited just if it has a point $1 \to [\![A]\!]$ in every Kripke model $(K, [\![-]\!])$.

(ii) Two terms are provably equal, $\mathbb{T} \vdash (\Gamma \mid s = t : A)$, just if they are equal in every Kripke model $(K, [\![-]\!])$,

$$[\![s]\!] = [\![t]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]\,.$$

For the proof, see [AR11], as well as [AGH21].

Remark 4.8.2. One can reformulate the Kripke semantics for simple type theory in terms of *discrete opfibrations* of posets,

$$\pi : F \longrightarrow K\,,$$

rather than (covariant) presheaves $F : K \to \mathsf{Set}$. Indeed, since the (full!) subcategory of all such maps

$$\mathsf{dopFib}/_K \hookrightarrow \mathsf{Pos}/_K$$

is equivalent to Set^K, this category is also cartesian closed. And with its obvious forgetful functor

$$\mathsf{dopFib}/_K \longrightarrow \mathsf{Pos}\,,$$

this provides another useful perspective on the functor category Set^K. This “fibrational” point of view is pursued in [AR11]. It is particularly useful for the semantics of *dependent* type theory, which we shall consider in next.

4.9 Dependent type theory

The Curry-Howard correspondence from Section 4.1 can be extended to natural deduction proofs in *first-order* logic, providing a refinement of the “propositions as types/proofs as terms” idea from propositional to first-order logic. In addition to simple types $A, B, ...$ representing propositions, one has *dependent* types $x : A \mid B(x)$ representing propositional functions. In addition to the simple type formers $A \times B$ and $A \to B$, one has dependent type formers $\Sigma_{x:A}B(x)$ and $\Pi_{x:A}B(x)$, representing the quantified formulas $\exists_{x:A}B(x)$ and $\forall_{x:A}B(x)$. As before, these types may have different terms $s, t : \Pi_{x:A}B(x)$, resulting from different proofs of the corresponding propositions, so that the calculus of terms records more information than mere *provability.* Also as before, the resulting structure turns out to be one that is shared by other categories *not* arising from logic—and now the coincidence is even more remarkable, because the structure at issue is a much richer and more elaborate one. Where proofs in the propositional calculus gave rise to a Cartesian closed category, the category of proof terms of first-order logic will be seen to be *locally Cartesian closed*, a mathematical structure also shared by sheaves on a space, Grothendieck toposes, categories of fibrations, and other important examples.

Recall first the notion of a hyperdoctrine $P : \mathcal{C}^{\mathsf{op}} \to \mathsf{Cat}$ from Section 3.4, and in particular the distinction between poset-valued and proper ones. The latter correspond more closely to dependent type theory, where the individual value categories $P(C)$ may be, e.g., cartesian closed, but they must also admit adjoints $\Sigma_A \dashv p_A^* \dashv \Pi_A$ along all projections $p_A : X \times A \to A$ in the category $\mathcal{C}$ of contexts. An important *difference* between hyperdoctrines and dependent type theories, however, is that the category of contexts in dependent type theory has not just finite products or finite limits, but also additional structure resulting from an operation of *context extension*, which takes as input a type in context $\Gamma \mid A$ and returns a new context $(\Gamma, x : A)$ together with a substitution $(\Gamma, x : A) \to \Gamma$. This is taking the "propositions-as-types" idea seriously, by allowing every proposition $\Gamma \mid \varphi$ in first-order logic to form a new type $\{\Gamma \mid \varphi\}$, or letting the objects $A \in P(C)$ in a hyperdoctrine $(\mathcal{C}, P)$ become arrows $\{A\} \to C$ in $\mathcal{C}$.[2]

Dependently-typed lambda-calculus. We give a somewhat informal specification of the syntax of the *dependently-typed λ-calculus* (see [**?**] for a more detailed exposition). To formulate the rules, we revisit the rules of simple type theory from section 4.3 and adjust them as follows.

Judgements: There are three kinds of judgements: for contexts, types, and terms, respectively,

$$\Gamma\ \mathtt{ctx}\,, \qquad \Gamma \mid A\ \mathtt{type}\,, \qquad \Gamma \mid a : A\,.$$

For each of these there are also (judgemental) equalities, the rules for which are the expected ones.

Contexts: These are formed by the rules:

$$\frac{}{(\cdot)\ \mathtt{ctx}} \qquad\qquad \frac{\Gamma \mid A\ \mathtt{type}}{\Gamma, x : A\ \mathtt{ctx}}$$

Here it is assumed that x is a fresh variable, not already occurring in Γ. Note that the order of the types occurring in a context matters, since types to the right may depend on ones to their left.

Types: In addition to the usual *simple types*, generated from *basic types* by formation of *products* and *function types*, we may also have some *basic types in context*,

$$\text{Basic dependent types} \quad \Gamma_1 \mid \mathtt{B}_1,\ \Gamma_2 \mid \mathtt{B}_2,\ \cdots$$

where the contexts Γ need not be basic. Further dependent types are formed from the basic ones by the Σ and Π type formers, using the rules:

$$\frac{\Gamma, x : A \mid B\ \mathtt{type}}{\Gamma \mid \Sigma_{x:A} B\ \mathtt{type}} \qquad\qquad \frac{\Gamma, x : A \mid B\ \mathtt{type}}{\Gamma \mid \Pi_{x:A} B\ \mathtt{type}}$$

[2] [Law70] does just this.

Terms: As for simple types, we assume there is a countable set of *variables* $x, y, z, \ldots$. We are also given a set of *basic constants*. The set of *terms* is generated from variables and basic constants by the following grammar, just as for simple types:

$$\begin{aligned} \text{Variables} \quad & v ::= x \mid y \mid z \mid \cdots \\ \text{Constants} \quad & c ::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\ \text{Terms} \quad & t ::= v \mid c \mid * \mid \langle t_1, t_2 \rangle \mid \mathtt{fst}\, t \mid \mathtt{snd}\, t \mid t_1\, t_2 \mid \lambda x : A \,.\, t \end{aligned}$$

The rules for deriving typing judgments are as for simple types:

- Each basic constant $\mathtt{c}_i$ has a uniquely determined type C_i (not necessarily basic):

  $$\frac{}{\Gamma \mid \mathtt{c}_i : C_i}$$

- The type of a variable is determined by the context:

  $$\frac{}{x_1 : A_1, \ldots, x_i : A_i, \ldots, x_n : A_n \mid x_i : A_i} \; (1 \leq i \leq n)$$

- The constant $*$ has type $\mathtt{1}$:

  $$\frac{}{\Gamma \mid * : \mathtt{1}}$$

- The typing rules for pairs and projections now take the form:

  $$\frac{\Gamma \mid a : A \qquad \Gamma \mid b : B(a)}{\Gamma \mid \langle a, b \rangle : \Sigma_{x:A} B} \qquad \frac{\Gamma \mid c : \Sigma_{x:A} B}{\Gamma \mid \mathtt{fst}\, c : A} \qquad \frac{\Gamma \mid c : \Sigma_{x:A} B}{\Gamma \mid \mathtt{snd}\, c : B(\mathtt{fst}\, c)}$$

  We write e.g. $B(a)$ rather than $B[a/x]$ to indicate a substitution of the term a for the variable x in the type B. We treat $A \times B$ as another way of writing $\Sigma_{x:A} B$, when the variable $x : A$ does not occur in the type B.

- The typing rules for application and λ-abstraction are now:

  $$\frac{\Gamma \mid t : \Pi_{x:A} B \qquad \Gamma \mid a : A}{\Gamma \mid t\, a : B(a)} \qquad \frac{\Gamma, x : A \mid t : B}{\Gamma \mid (\lambda x : A \,.\, t) : \Pi_{x:A} B}$$

  We treat $A \to B$ as another way of writing $\Pi_{x:A} B$, when the variable $x : A$ does not occur in the type B.

Equations: The (β and η) equations between these terms are just as they were for simple types. There are also the usual equations making judgemental equality a congruence with respect to all type and term formers.

Equality types: Just as for first-order logic, we may also add a primitive *equality type* $x =_A y$ for each type A, sometimes called *propositional equality*, and not to be confused with the judgemental equality, which we shall now write as $s \equiv t$ to emphasize the difference. The formation, introduction, elimination, and computation rules for equality types are as follows:

$$\frac{\Gamma \mid s : A \qquad \Gamma \mid t : A}{\Gamma \mid s =_A t \ \mathtt{type}} \qquad\qquad \frac{\Gamma \mid a : A}{\Gamma \mid \mathtt{refl}_a : (a =_A a)}$$

$$\frac{\Gamma \mid p : s =_A t}{\Gamma \mid s \equiv t : A} \qquad\qquad \frac{\Gamma \mid p : s =_A t}{\Gamma \mid p \equiv \mathtt{refl}_s : (s =_A s)}$$

Remark 4.9.1 (Identity types)**.** This formulation of the rules for equality is known as the *extensional* theory. There is also an *intensional* version, with different elimination (and computation) rules, in which the types are sometimes called *identity types* and written $\mathtt{Id}_A(s,t)$ instead. See [**?**] for details.

Example 4.9.2 (The type-theoretic axiom of choice)**.** Reading Σ as "there exists" and Π as "for all", a type such a $\Pi_{x:A}\Sigma_{y:B}R(x,y)$ can be regarded as a stating a proposition—in this case, "for all $x : A$ there is a $y : B$ such that $R(x,y)$". By Curry-Howard, such a "proposition" is then provable if it has a closed term $t : \Pi_{x:A}\Sigma_{y:B}R(x,y)$, which then corresponds to a proof, by unwinding the rules that constructed the term, and observing that they correspond to the usual natural deduction rules for first-order logic.

This only partly true, however: the rules of construction for terms correspond to provability under a certain "constructive" conception of validity (see [**?**]). This is made clear by the following example, which is sometimes called the "type theoretic axiom of choice", because it sounds like the axiom of choice under the conventional interpretation; but this statement is actually provable from the rules of type theory, rather than being an axiom!

$$\Pi_{x:A}\Sigma_{y:B}R(x,y) \to \Sigma_{f:A\to B}\Pi_{x:A}R(x,fx)\,. \tag{4.9}$$

Exercise 4.9.3. Prove the type theoretic axiom of choice (4.9) from the rules for dependent type theory given here.

4.10 Locally cartesian closed categories

Recall the following from Proposition 3.3.3.

Proposition 4.10.1. *The following conditions on a category $\mathcal{C}$ with terminal object 1 are equivalent:*

1. *Every slice category $\mathcal{C}/A$ is cartesian closed.*

2. *For every arrow $f : B \to A$, the (post-) composition functor $\Sigma_f : \mathcal{C}/B \to \mathcal{C}/A$ has a right adjoint f^*, which in turn has a right adjoint Π_f,*

$$\begin{tikzcd}
B \arrow[r, "f"] & A
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{C}/B \arrow[r, bend left, "\Sigma_f"] \arrow[r, bend right, "\Pi_f"'] & \mathcal{C}/A \arrow[l, "f^*" description]
\end{tikzcd}$$

Such a category is called locally cartesian closed*.*

The notation of course anticipates the interpretation of DTT.

Proof. Construct Π from exponentials and pullbacks; see the proof of Proposition 3.3.3. □

Basic examples of LCCCs

We have the following basic examples, most of which we have already seen in Section 3.4 on hyperdoctrines,

1. Set: We have already seen the hyperdoctrine Set^I of *families of sets* $(A_i)_{i\in I}$, with action of $f : J \to I$ on $A : I \to \mathsf{Set}$ by precomposition $f^*A = A \circ f : J \to \mathsf{Set}$. The equivalent hyperdoctrine
$$\mathsf{Set}^I \ \simeq\ \mathsf{Set}/_I$$
uses the slice categories $\mathsf{Set}/_I$ with action by pullback $f^* : \mathsf{Set}/_I \to \mathsf{Set}/_J$. It follows that Set is locally cartesian closed.

2. Presheaves: The LCC structure on presheaves $\widehat{\mathbb{C}}$ on a small category $\mathbb{C}$ follows from the CCC structure on each slice, since each of the slice categories $\widehat{\mathbb{C}}/_X$ is another category of presheaves, namely $\widehat{\int_{\mathbb{C}} X}$, on the category of elements $\int_{\mathbb{C}} X$. That $\widehat{\mathbb{C}}$ is a CCC is shown directly by computing the products of presheaves $P \times Q$ pointwise, and the exponential as $Q^P = \mathsf{Hom}(\mathsf{y}(-) \times P, Q)$.

3. Pos: The category of posets is cartesian closed, but not locally so. However, we have seen that the category of discrete fibartions on a poset K is equivalent to a category of presheaves $\mathsf{dFib}(K) \simeq \mathsf{Set}^{K^{\mathsf{op}}}$. It follows that the (non-full) subcategory $\mathsf{dFib} \hookrightarrow \mathsf{Pos}$ of posets and discrete fibrations as arrows would be locally cartesian closed *except* for the fact that it lacks a terminal object. Thus every slice of this category $\mathsf{dFib}(K)/P \simeq \mathsf{Set}^{K^{\mathsf{op}}}/P \simeq \mathsf{Set}^{(\int_K P)^{\mathsf{op}}}$ *is* LCC.

4. An example similar to the foregoing is the non-full subcategory $\mathsf{LocHom} \hookrightarrow \mathsf{Top}$ of topological spaces and local homeomorphisms between them, which lacks a terminal object, but each slice of which $\mathsf{LocHom}/X \simeq \mathsf{Sh}(X)$ is equivalent to the topos of *sheaves* on the space X, and is therefore CCC (and so LCCC).

Exercise 4.10.2. Let $P : \mathcal{C}^{\mathsf{op}} \to \mathsf{Cat}$ be a hyperdoctrine for which there are equivalences $PC \simeq \mathcal{C}/C$, naturally in C, with respect to the left adjoints $\Sigma_f : \mathcal{C}/A \to \mathcal{C}/B$ for all $f : A \to B$ in $\mathcal{C}$. Show that $\mathcal{C}$ is then LCC.

Exercise 4.10.3. Show that any LCCC $\mathcal{C}$, regarded as a hyperdoctrine, has equality in the sense of Remark 3.4.3.

4.11 Functorial semantics of DTT in LCCCs

In the semantics of dependent type theory in a locally cartesian closed category, contexts are interpreted as objects, and dependent types as morphisms. Let $\mathcal{C}$ be an LCCC and interpret the empty context as the terminal object, $\llbracket \cdot \rrbracket = 1$, and for a closed type $\cdot \mid B$, let $\llbracket \cdot \mid B \rrbracket : \llbracket B \rrbracket \to 1$. More generally, given any type in context $\Gamma \mid A$, we shall have

$$\llbracket \Gamma \mid A \rrbracket : \llbracket \Gamma, A \rrbracket \longrightarrow \llbracket \Gamma \rrbracket ,$$

abbreviating $\Gamma, x : A$ to Γ, A. Specifically, given $\Gamma, A \mid B$, we then have maps

$$\llbracket \Gamma, A, B \rrbracket \overset{\llbracket \Gamma, A \mid B \rrbracket}{\longrightarrow} \llbracket \Gamma, A \rrbracket \overset{\llbracket \Gamma \mid A \rrbracket}{\longrightarrow} \llbracket \Gamma \rrbracket ,$$

and we use the left and right adjoints to pullback to interpret the eponymous type-forming operations:

$$\begin{aligned} \llbracket \Gamma \mid \Sigma_{x:A} B \rrbracket &= \Sigma_{\llbracket \Gamma \mid A \rrbracket}(\llbracket \Gamma, A \mid B \rrbracket) , \\ \llbracket \Gamma \mid \Pi_{x:A} B \rrbracket &= \Pi_{\llbracket \Gamma \mid A \rrbracket}(\llbracket \Gamma, A \mid B \rrbracket) . \end{aligned}$$

A term $\Gamma \mid a : A$ is interpreted as a section:

$$\begin{tikzcd} \llbracket \Gamma \rrbracket \arrow[r, "\llbracket \Gamma \mid a : A \rrbracket"] \arrow[dr, "="'] & \llbracket \Gamma, A \rrbracket \arrow[d, "\llbracket \Gamma \mid A \rrbracket"] \\ & \llbracket \Gamma \rrbracket \end{tikzcd}$$

Finally, as in first-order logic, substitution of a term $\Gamma \mid a : A$ for a variable $\Gamma, x : A$ is interpreted by pullback,

$$\begin{tikzcd} \llbracket \Gamma, B(a) \rrbracket \arrow[r] \arrow[d, "\llbracket \Gamma \mid B(a) \rrbracket"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \llbracket \Gamma, A, B \rrbracket \arrow[d, "\llbracket \Gamma, A \mid B \rrbracket"] \\ \llbracket \Gamma \rrbracket \arrow[r, "\llbracket \Gamma \mid a : A \rrbracket"'] & \llbracket \Gamma, A \rrbracket \end{tikzcd}$$

and similarly for substitution into terms. The interpretation of substitution as pullback leads to a coherence problem that we shall consider in the next section.

As was done for simple type theory in Section 4.6, we can again develop the relationship between the type theory and its models using the framework of *functorial semantics*. This is now a common generalization of λ-theories, modeled in CCCs, and first-order logic, modeled in Heyting categories. The first step is to build a syntactic *classifying category* $\mathcal{C}_{\mathbb{T}}$ from a theory $\mathbb{T}$ in dependent type theory, which we then show classifies $\mathbb{T}$-models in LCCCs. We omit the now essentially routine details (given the analogous cases already considered), and merely state the main result, the proof of which is also analogous to the previous cases. A detailed treatment can be found in the seminal paper [See84].

Theorem 4.11.1. *For any theory $\mathbb{T}$ in dependent type theory, the locally cartesian closed syntactic category $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models, in the sense that for any locally cartesian closed category $\mathcal{C}$ there is an equivalence of categories*

$$\mathsf{Mod}\big(\mathbb{T},\mathcal{C}\big)^{i} \;\simeq\; \mathsf{LCCC}\big(\mathcal{C}_{\mathbb{T}},\mathcal{C}\big)^{i}, \tag{4.10}$$

naturally in $\mathcal{C}$. The morphisms of $\mathbb{T}$-models on the left are the isomorphisms of the underlying structures, and on the right we take the natural isomorphisms of LCCC functors.

As a corollary, again as before, we have that dependent type theory is *complete* with respect to the semantics in locally cartesian closed categories, in virtue of the syntactic construction of the classifying category $\mathcal{C}_{\mathbb{T}}$. Specifically, any theory $\mathbb{T}$ has a canonical interpretation $[-]$ in the syntactic category $\mathcal{C}_{\mathbb{T}}$ which is *logically generic* in the sense that, for any terms $\Gamma \mid s : A$ and $\Gamma \mid t : A$, we have

$$\begin{aligned} \mathbb{T} \vdash (\Gamma \mid u \equiv t : A) \quad &\Longleftrightarrow \quad [\Gamma \mid u : A] = [\Gamma \mid t : A] \\ &\Longleftrightarrow \quad [-] \models (\Gamma \mid s \equiv t : A)\,. \end{aligned}$$

Thus, for the record, we have:

Proposition 4.11.2. *For any dependently typed theory $\mathbb{T}$,*

$$\mathbb{T} \vdash (\Gamma \mid u \equiv t : A) \quad \textit{if, and only if,} \quad \mathcal{C}_{\mathbb{T}} \models (\Gamma \mid u \equiv t : A)\,.$$

Of course, the syntactic model $[-]$ in $\mathcal{C}_{\mathbb{T}}$ is the one associated under (4.10) to the identity functor $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}}$, *i.e.* it is the *universal* one. It therefore satisfies an equation just in case the equation holds in *all* models, by the classifying property of $\mathcal{C}_{\mathbb{T}}$, and the preservation of satisfaction of equations by LCCC functors (as in Proposition 4.4.2).

Corollary 4.11.3. *For any dependently typed theory $\mathbb{T}$,*

$$\mathbb{T} \vdash (\Gamma \mid u \equiv t : A) \quad \textit{if, and only if,} \quad M \models (\Gamma \mid u \equiv t : A) \textit{ for every LCCC model } M.$$

Moreover, a closed type A is inhabited $\vdash a : A$ if, and only if, there is a point $1 \to [\![A]\!]^M$ in every model M.

The embedding and completeness theorems of Section 4.7 with respect to general presheaf and Kripke models can also be extended to dependently typed theories. See [AR11] for details. There is also a version of Kripke-Joyal forcing for such theories (and an associated completeness theorem), for which the interested reader can consult [AGH21].

4.12 Coherence and natural models

The semantics of DTT in LCCCs described in the previous section uses the “slice category” hyperdoctrine of an LCC to interpret the dependent types. Thus the contexts Γ and substitutions $\sigma : \Delta \to \Gamma$ are interpreted as the objects and arrow of an LCC category $\mathcal{C}$, and the dependent types $\Gamma \mid A$ and terms $\Gamma \mid a : A$ are interpreted as objects $A \to \Gamma$ in the slice category $\mathcal{C}/\Gamma$ and their global sections $a : \Gamma \to A$ (over Γ). However, there is a problem with this kind of semantics (as first pointed out by [Hof]): as a hyperdoctrine, this interpretation is a *pseudo*functor $\mathcal{C}/ : \mathcal{C}^{\mathsf{op}} \to \mathsf{Cat}$, but the syntax of DTT produces an actual *presheaf* of types in context $\mathsf{Ty} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$, since substitution into dependent types is *strictly* functorial with respect to composition of substitutions, in the sense that for a type in context $\Gamma \mid A$ and substitutions $\sigma : \Delta \to \Gamma$ and $\tau : \Theta \to \Delta$ we have an *equality of types in context*,

$$\Theta \mid (A[\sigma])[\tau] \equiv A[\sigma \circ \tau]\,,$$

rather than the (canonical) isomorphism $\cong$ fitting into the two-pullbacks diagram of the hyperdoctrine, namely:

$$\begin{tikzcd}
(\sigma \circ \tau)^* A \arrow[r, "\cong"'] \arrow[rrr, bend left] \arrow[dr] & \tau^*\sigma^* A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \sigma^* A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d] \\
 & \Theta \arrow[r, "\tau"] \arrow[rr, bend right, "\sigma \circ \tau"'] & \Delta \arrow[r, "\sigma"] & \Gamma
\end{tikzcd}$$

A similar problem occurs in the Beck-Chavalley conditions, where the hyperdoctrine structure has only canonical isos, rather than the strict equalities that obtain in the syntax, such as

$$(\Pi_{x:A} B)[\sigma] \equiv (\Pi_{x:A[\sigma]} B[\sigma])\,.$$

There are various different solutions to this problem in the literature, some involving “strictifications” of the LCC slice-category hyperdoctrine (including both left- and right-adjoint strictifications), as well as other semantics altogether, such as categories-with-families [Dyb96], categories-with-attributes, and comprehension categories.

A solution based on the notion of *universe* $\widetilde{U} \to U$ was first proposed by Voevodsky; this approach is combined with the notion of a *representable natural transformation* in [Awo16] as follows.

Definition 4.12.1. For a small category $\mathbb{C}$, a natural transformation $f : Y \to X$ of presheaves on $\mathbb{C}$ is called *representable* if for every $C \in \mathbb{C}$ and $x \in X(C)$, there is given a

$D \in \mathbb{C}$, a $p : D \to C$, and a $y \in Y(D)$ such that the following square is a pullback.

$$\begin{tikzcd} \mathsf{y}D \arrow[r, "y"] \arrow[d, "\mathsf{y}p"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "f"] \\ \mathsf{y}C \arrow[r, "x"'] & X \end{tikzcd} \tag{4.11}$$

A representable natural transformation is the same thing as a *category with families* in the sense of Dybjer [Dyb96]. Indeed, let us write the objects of $\mathbb{C}$ as $\Gamma, \Delta, \ldots$ and the arrows as $\sigma : \Delta \to \Gamma, \ldots$, thinking of $\mathbb{C}$ as a "category of contexts". Let $p : E \to U$ be a representable map of presheaves, and write its elements as:

$$\begin{aligned} A \in U(\Gamma) &\quad \text{iff} \quad \Gamma \mid A \\ a \in E(\Gamma) &\quad \text{iff} \quad \Gamma \mid a : A, \end{aligned}$$

where $A = p \circ a$, as indicated in:

$$\begin{tikzcd} & E \arrow[d, "p"] \\ \mathsf{y}\Gamma \arrow[ur, "a"] \arrow[r, "A"'] & U. \end{tikzcd}$$

Thus we regard U as the *presheaf of types*, with $U(\Gamma)$ the set of all types in context Γ, and E as the *presheaf of terms*, with $E(\Gamma)$ the set of all terms in context Γ, while the component $p_\Gamma : E(\Gamma) \to U(\Gamma)$ is the typing of the terms in context Γ.

Naturality of $p : E \to U$ just means that for any substitution $\sigma : \Delta \to \Gamma$, we have an action on types and terms:

$$\begin{aligned} \Gamma \mid A &\quad \mapsto \quad \Delta \mid A\sigma \\ \Gamma \mid a : A &\quad \mapsto \quad \Delta \mid a\sigma : A\sigma\,. \end{aligned}$$

While, by functoriality, given any further $\tau : \Theta \to \Delta$, we have

$$(A\sigma)\tau = A(\sigma \circ \tau) \qquad (a\sigma)\tau = a(\sigma \circ \tau),$$

as well as

$$A1 = A \qquad a1 = a$$

for the identity substitution $1 : \Gamma \to \Gamma$.

Finally, the representability of the natural transformation $p : E \to U$ is exactly the operation of *context extension*: given any $\Gamma \mid A$, by Yoneda we have the corresponding map $A : \mathsf{y}\Gamma \to U$, and we let $p_A : \Gamma.A \to \Gamma$ be (the map representing) the pullback of p along A, as in (4.11). We therefore have a pullback square:

$$\begin{tikzcd}
\mathsf{y}\Gamma.A \arrow[r, "q_A"] \arrow[d, "\mathsf{y}p_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & E \arrow[d, "p"] \\
\mathsf{y}\Gamma \arrow[r, "A"'] & U,
\end{tikzcd} \tag{4.12}$$

where the map $q_A : \Gamma.A \to E$ now determines a term

$$\Gamma.A \mid q_A : Ap_A.$$

We may omit the y for the Yoneda embedding, letting the Greek letters serve to distinguish representable presheaves.

Exercise 4.12.2. Show that the fact that (4.12) is a pullback means that given any $\sigma : \Delta \to \Gamma$ and $\Delta \mid a : A\sigma$, there is a map

$$(\sigma, a) : \Delta \to \Gamma.A,$$

and this operation satisfies the equations

$$\begin{aligned} p_A \circ (\sigma, a) &= \sigma \\ q_A(\sigma, a) &= a, \end{aligned}$$

as indicated in the following diagram.

$$\begin{tikzcd}
\Delta \arrow[dr, dotted, "{(\sigma, a)}"] \arrow[drr, bend left, "a"] \arrow[ddr, bend right, "\sigma"'] & & \\
& \Gamma.A \arrow[r, "q_A"'] \arrow[d, "p_A"] \arrow[dr, phantom, "\lrcorner", very near start] & E \arrow[d, "p"] \\
& \Gamma \arrow[r, "A"'] & U
\end{tikzcd}$$

Show moreover that the uniqueness of (σ, a) means that for any $\tau : \Delta' \to \Delta$ we also have:

$$\begin{aligned} (\sigma, a) \circ \tau &= (\sigma \circ \tau, a\tau) \\ (p_A, q_A) &= 1. \end{aligned}$$

Comparing the foregoing with the definition of a category with families in [Dyb96], we have shown:

Proposition 4.12.3. *Let $p : E \to U$ be a natural transformation of presheaves on a small category $\mathbb{C}$ with a terminal object. Then p is representable in the sense of Definition 4.12.1 just in case it determines a category with families, as just indicated.*

The notion of a category with families is a variable-free way of presenting dependent type theory, including contexts and substitutions, types and terms in context, and context extension. Accordingly, we may think of a representable map of presheaves on a category $\mathbb{C}$ as a "type theory over $\mathbb{C}$" as the category of contexts and substitutions. (This is the reason for the requirement that $\mathbb{C}$ should have a terminal object to represent the "empty context".) One can also show that such a map of presheaves is essentially determined by a class of maps in $\mathbb{C}$ that is closed under all pullbacks, corresponding to the types in context (see [Awo16]).

Definition 4.12.4. A *natural model of type theory* on a small category $\mathbb{C}$ is a representable map of presheaves $p : E \to U$.

Exercise 4.12.5. Let $\mathbb{T}$ be a dependent type theory and $\mathcal{C}_{\mathbb{T}}$ its category of contexts and substitutions. Define the presheaves $\mathsf{Ty} : {\mathcal{C}_{\mathbb{T}}}^{\mathsf{op}} \to \mathsf{Set}$ of types-in-context and $\mathsf{Tm} : {\mathcal{C}_{\mathbb{T}}}^{\mathsf{op}} \to \mathsf{Set}$ of terms-in-context, along with a natural transformation,

$$\tau : \mathsf{Tm} \to \mathsf{Ty}$$

that takes a term to its type. Show that $\tau : \mathsf{Tm} \to \mathsf{Ty}$ is a natural model of type theory.

4.13 Universes

4.14 Induction and W-types

Chapter 5

Dependent Type Theory

5.1 Dependent type theory

5.2 Inductive types

5.2.1 Initial algebras for endofunctors

Construction by iteration.

Induction principles.

Examples:

- lifting functor in domain theory

5.2.2 Inductive and coinductive types

NNO as inductive type.

Induction and coinduction.

Inductive and coinductive types for polynomial functors, perhaps for more general functors.

Existence of W-types in toposes.

Other possible examples of W-types:

- syntax: the expressions of a language are a W-type for a suitable signature
- data structures: lists, trees, etc.
- lazy data structures: lazy lists, lazy numbers, regular infinite trees, streams (these are coinductive).
- Von Neumann ordinals as well-founded infinitely branching trees.

5.2.3 Bracket types

5.3 Dependent type theory with FOL

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:

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

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

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

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:

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

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

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

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

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

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 $\Gamma(f) = \big\{\langle x, f(x)\rangle \;\big|\; x \in A\big\}$ 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:

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

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

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

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

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

[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

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

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:

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

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

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,

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

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 $\eta_A = \theta_{A,FA} 1_{FA} : A \to G(FA)$,

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

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,

$$FX = \big\{ x_1 \dots x_n \bigm| n \geq 0 \ \& \ x_1, \dots, x_n \in X \big\} \ .$$

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:

$$\frac{1_{GB} : GB \to GB}{\varepsilon_B : F(GB) \to B}$$

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

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

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:

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

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 $q_1 = p_1 \circ h$:

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

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:

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

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

$$\left\{A_i \in \mathcal{C} \;\middle|\; i \in I\right\} \ .$$

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 $\left\{f_i : B \to A_i \;\middle|\; i \in I\right\}$ 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:

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

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 $r_1 = s \circ q_1$:

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

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 × 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.

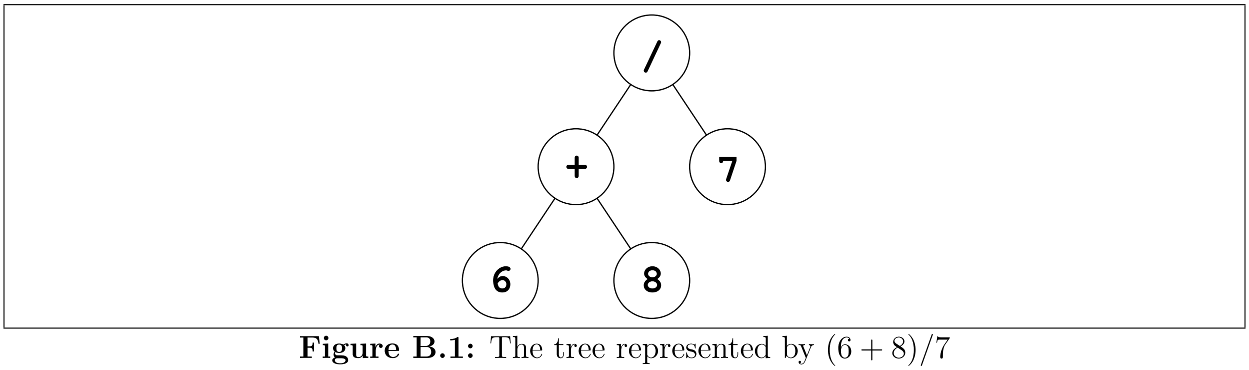


Figure B.1: The tree represented by $(6 + 8)/7$

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:

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

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.

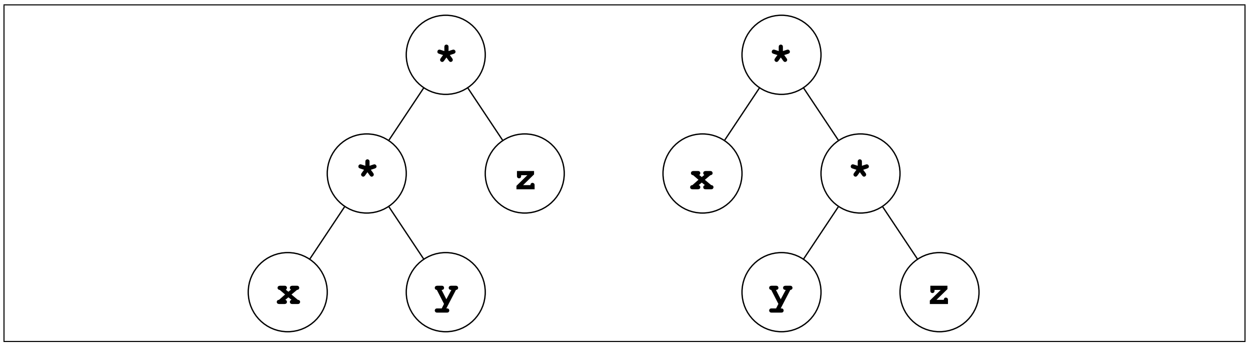


Figure B.2: Left and right associativity of $x \star y \star z$

[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

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

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 $\phi(x) \vee \exists\, x\,.\, A\psi(x, x)$ 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 $\phi(x) \vee \exists\, x\,.\, A\psi(x, x)$ but rather $\phi(x) \vee \exists\, y\,.\, A\psi(y, y)$. 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:

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

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.*

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

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

The equational calculus has just one form of judgement

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

where $x_1, \dots, x_n$ 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.* We write

$$\mathbb{T} \vdash t_1 = t_2$$

to mean that the equation $t_1 = t_2$ has a derivation from the axioms of $\mathbb{T}$ using the equational calculus.

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:

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

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.

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

$$\genfrac{}{}{}{}{\Phi \vdash \phi_1 \quad \Phi \vdash \phi_2}{\overline{\Phi \vdash \phi_1 \land \phi_2}} \qquad \genfrac{}{}{}{}{\Phi, \phi_1 \vdash \psi \quad \Phi, \phi_2 \vdash \psi}{\overline{\Phi, \phi_1 \lor \phi_2 \vdash \psi}} \qquad \genfrac{}{}{}{}{\Phi, \phi_1 \vdash \phi_2}{\overline{\Phi \vdash \phi_1 \Rightarrow \phi_2}}$$

$$\genfrac{}{}{}{}{\Gamma, x, y \mid \Phi, x = y \vdash \phi}{\overline{\Gamma, x \mid \Phi \vdash \phi[x/y]}} \qquad \genfrac{}{}{}{}{\Gamma, x \mid \Phi, \phi \vdash \psi}{\overline{\Gamma \mid \Phi, \exists x \,.\, \phi \vdash \psi}} \qquad \genfrac{}{}{}{}{\Gamma, x \mid \Phi \vdash \phi}{\overline{\Gamma \mid \Phi \vdash \forall x \,.\, \phi}}$$

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,

$$\frac{}{\Phi \vdash \phi \lor \neg\phi}$$

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

[AF13] S. Awodey and H. Forssell. First-order logical duality. *Annals of Pure and Applied Logic*, 164(3):319–348, 2013.

[AGH21] S. Awodey, N. Gambino, and S. Hazratpour. Kripke-Joyal forcing for type theory and uniform fibrations, October 2021. Preprint available as `https://arxiv.org/abs/2110.14576`.

[ALR03] J. Adamek, F. W. Lawvere, and J. Rosiky. On the duality between varieties and algebraic theories. *Algebra Universalis*, 49:35–49, 2003.

[AR94] Jiri Adamek and Jiri Rosicky. *Locally Presentable and Accessible Categories.* Number 189 in London Mathematical Society Lecture Notes. Cambridge University Press, 1994.

[AR11] S. Awodey and F. Rabe. Kripke semantics for Martin-Löf's extensional type theory. *Logical Methods in Computer Science*, 7(3):1–25, 2011.

[ARV10] J. Adamek, J. Rosicky, and E.M. Vitale. *Algebraic Theories.* Cambridge University Press, 2010.

[Awo10] Steve Awodey. *Category Theory.* Number 52 in Oxford Logic Guides. Oxford University Press, 2010.

[Awo16] Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28:1–46, 11 2016.

[Awo21] Steve Awodey. Sheaf representations and duality in logic. In C. Casadio and P.J. Scott, editors, *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics.* Springer, 2021. arXiv:2001.09195.

[Bir35] Garrett Birkhoff. On the structure of abstract algebras. *Proc. Camb. Philos. Soc.*, 31:433–454, 1935.

[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.

[Dyb96] P. Dybjer. Internal type theory. *LNCS*, 1158:120–134, 1996.

[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.

[Hof] Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and logics of computation.*

[HS98] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, New York, 1998.

[Joh82] P.T. Johnstone. *Stone Spaces.* Number 3 in Cambridge studies in advanced mathematics. Cambridge University Press, 1982.

[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.

[Mak87] Michael Makkai. Stone duality for first order logic. *Adv. Math.*, 65:97–170, 1987.

[Mak93] Michael Makkai. *Duality and Definability*, volume 503. AMS, 1993.

[McC93] W. McCune. Single axioms for groups and abelian groups with various operations. *Journal of Automated Reasoning*, 10(1):1–13, 1993.

[MH92] Michael Makkai and Victor Harnik. Lambek's categorical proof theory and Läuchli's abstract realizability. *Journal of Symbolic Logic*, 57(1):200–230, 1992.

[ML84] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory.* Bibliopolis, 1984.

[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.

[Pal03] Erik Palmgren. Groupoids and local cartesian closure. 08 2003. unpublished.

[PV07] E. Palmgren and S.J. Vickers. Partial horn logic and cartesian categories. *Annals of Pure and Applied Logic*, 145(3):314–353, 2007.

[Sco70] Dana S. Scott. Constructive validity. In M. Laudet, D. Lacombe, L. Nolin, and M. Schützenberger, editors, *Symposium on Automatic Demonstration*, volume 125, pages 237–275. Springer-Verlag, 1970.

[Sco80] Dana S. Scott. Relating theories of the lambda calculus. In J. Roger Seldin, Jonathan P.; Hindley, editor, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 403–450. Academic Press, 1980.

[See84] R. A. G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(1):33, 1984.

[Tai68] William W. Tait. Constructive reasoning. In *Logic, Methodology and Philos. Sci. III (Proc. Third Internat. Congr., Amsterdam, 1967)*, pages 185–199. North-Holland, Amsterdam, 1968.