Categorical Logic

Autumn School on Proof and Computation

Steve Awodey
(with input from Andrej Bauer)

Fischbachau, Germany
26.9.–1.10.2022

Contents

Chapter 1

Category Theory

1.1 Categories

Definition 1.1.1. 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:

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

- 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 1.1.2. 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.

1.1.1 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 $\mathbf{1}_\star$:

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

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

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

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

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

The composition of arrows is given by the group operation:

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

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

We can turn things around and *define* a monoid to be a category with a single object. A group is then a category with a single object in which every arrow is an *isomorphism* (in the sense of definition 1.1.5 below).

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

$$x \leq x \qquad \text{(reflexive)}$$
$$x \leq y \;\&\; y \leq z \;\Rightarrow\; x \leq z \qquad \text{(transitive)}$$
$$x \leq y \;\&\; y \leq x \;\Rightarrow\; x = y \qquad \text{(antisymmetric)}$$

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*.

1.1.2 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 1.1.3. 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:

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

Show that this is indeed a category!

1.1.3 Basic notions

We recall some further basic notions from category theory.

Definition 1.1.4. 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 1.1.5. 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 1.1.6. 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*.

1.2 Functors

Definition 1.2.1. 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 1.2.2. 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.

1.2.1 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 1.2.3. Verify that the above claims are correct.

1.2.2 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 1.2.4. Show that taking the graph $\Gamma(f) = \big\{\langle x, f(x)\rangle \bigm| x \in A\big\}$ 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?

1.3 Constructions of Categories and Functors

1.3.1 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:

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

Exercise 1.3.1. Show that, for any categories $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, there are distinguished isos:

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

Does this make Cat a (commutative) monoid?

1.3.2 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,

$$\begin{tikzcd} B \arrow[d, "f"] \\ A \end{tikzcd} \tag{1.1}$$

and as morphisms, commutative diagrams over A:

$$\begin{tikzcd} B \arrow[rr, "g"] \arrow[dr, "f"'] & & B' \arrow[dl, "f'"] \\ & A & \end{tikzcd} \tag{1.2}$$

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 (1.1) to its domain B, and a morphism (1.2) 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 (1.1) to the object $h \circ f : B \to A'$ and a morphisms (1.2) 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

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

1.3.3 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 $\mathcal{C}$,

$$\begin{tikzcd} A \arrow[d, "f"] \\ B \end{tikzcd} \tag{1.3}$$

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

$$\begin{tikzcd} A \arrow[r, "g"] \arrow[d, "f"'] & A' \arrow[d, "f'"] \\ B \arrow[r, "g'"'] & B'. \end{tikzcd} \tag{1.4}$$

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?)

1.3.4 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 1.3.2. 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?

1.3.5 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

$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D \tag{1.5}$$

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 (1.5) 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?)

1.3.6 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 1.3.3. 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.

1.4 Natural Transformations and Functor Categories

Definition 1.4.1. 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 1.4.2. 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

$$\mathsf{Cat}(\mathbb{A} \times \mathbb{B}, \mathbb{C}) \cong \mathsf{Cat}(\mathbb{A}, \mathbb{C}^{\mathbb{B}}) \ . \tag{1.6}$$

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:

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

1.4.1 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 1.4.3. Exhibit the arrow category $\mathcal{C}^{\to}$ and the category of group actions $\mathsf{Set}(G)$ as functor categories.

1.4.2 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 (1.6) 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 1.4.4. Show that this *is* a functor.

Theorem 1.4.5 (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 1.4.6 (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 1.4.5. 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 1.4.7. *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 1.4.8. 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 $\widehat{\mathcal{C}} = \mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$.

1.4.3 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 1.4.9. 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 1.4.10. *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

$$F(Gf) = {\eta_C}^{-1} \circ f \circ \eta_B \ . \tag{1.7}$$

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, (1.7) 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. $\square$

Example 1.4.11. 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

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

[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 1.4.12. 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 1.4.10, 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.

1.5 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.

1.5.1 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 $x \in P$, $y \in Q$,

$$
fx \leq y \iff x \leq gy \; . \tag{1.8}
$$

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 (1.8) is more conveniently displayed as

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

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:

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

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

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.

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

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.

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

Therefore, *conjunction is left adjoint to implication.*

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

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

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

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

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

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

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

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

1.5.2 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 1.5.1. 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 (1.6),

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

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 1.5.2. Verify that the definition (1.8) of adjoint monotone maps between preorders is a special case of Definition 1.5.1. 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

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$$

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 1.5.3. *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.

1.5.3 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 1.5.4. *A functor $F : \mathcal{C} \to \mathcal{D}$ is left adjoint to a functor $G : \mathcal{D} \to \mathcal{C}$ if, and only if, there exists a natural transformation*

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

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

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

is an isomorphism.

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

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

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

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

The monoid operation is concatenation of sequences

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

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

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

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

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

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

This is precisely the condition required by Proposition 1.5.4 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.

1.5.4 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 1.5.4.

Proposition 1.5.5. *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

$$\varepsilon_M(x_1 x_2 \dots x_n) = x_1 \star x_2 \star \dots \star x_n \ .$$

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.

1.6 Limits and Colimits

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

1.6.1 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.

1.6.2 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 1.6.1. Prove that if $\mathsf{1}$ and $\mathsf{1}'$ are terminal objects in a category then they are isomorphic.

Exercise 1.6.2. 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?

1.6.3 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 1.6.11.

[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 $k = e \circ m$:

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

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 1.6.3. 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 1.6.4. 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]

1.6.4 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 $q_1 = p_1 \circ h$:

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

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

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

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 $\{\langle x, y\rangle \in A \times U \mid fx = y\} \cong \{x \in A \mid fx \in U\} = f^*U$, the inverse image of U along f.

Exercise 1.6.5. 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:

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

1.6.5 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

$$\begin{tikzcd} & D_1 \arrow[dl, "d_{12}"'] \arrow[d, "d_{13}"] \\ D_2 \arrow[r, "d_{23}"'] & D_3 \end{tikzcd} \tag{1.9}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Exercise 1.6.6. 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 1.6.7. *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 $!_B$:

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

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 $!_A \circ f = !_B \circ g$. $\square$

Proposition 1.6.8. *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 $\Sigma = \phi \circ \Lambda$.

1.6.6 Colimits

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

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

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

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

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

Colimits are also called *inductive limits*.

Exercise 1.6.9. Formulate the dual of Proposition 1.6.7 and Proposition 1.6.8 for colimits (coequalizers are defined in Section 1.6.9).

1.6.7 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:

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

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 1.6.10. Show that the categories of posets and of topological spaces both have coproducts.

1.6.8 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 1.6.11. 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 1.6.12. 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 1.6.13. 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 $x \cdot y = y \cdot x$.

1.6.9 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 1.6.14. 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 1.6.3.]

Definition 1.6.15. 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]

1.6.10 Pushouts

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

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

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

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

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.

1.6.11 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:

$$\frac{!_A : A \to 1_{\mathcal{C}}}{\overline{\mathsf{1}_\star : !_{\mathcal{C}} A \to \star}}$$

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

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

Now consider the diagonal functor,

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

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

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

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

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

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

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

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

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,

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

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 1.6.16. 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?

1.6.12 Preservation of limits

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

$$A \xleftarrow{\pi_0} A \times B \xrightarrow{\pi_1} B$$

its image in $\mathcal{D}$,

$$FA \xleftarrow{F\pi_0} F(A \times B) \xrightarrow{F\pi_1} FB$$

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 1.6.17. *(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 1.6.8, and that colimits are constructed from coequalizers and coproducts, cf. Exercise 1.6.9. □

Proposition 1.6.18. *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. $\square$

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 1.6.19. 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 1.6.20. 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 1.6.5.

Proposition 1.6.21. *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. □

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 the *classical* propositional logic, before considering the intuitionistic case in Section 3.4.

In the style of Section **??**, we have the following (abstract) syntax for (propositional) formulas:

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

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

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

where $\mathtt{p}_1, \ldots, \mathtt{p}_n$ is a *context* consisting of distinct propositional variables, the formulas $\phi_1, \ldots, \phi_m$ 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 $\Phi \vdash \phi$ 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 a hypothesis.

Exercise 2.1.1. 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, and not depedent on 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 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 $\llbracket \phi \rrbracket^v = 1$. Then the above specification takes the form:

$$
\begin{array}{rll}
v \vDash \top & \text{always} & \\
v \vDash \bot & \text{never} & \\
v \vDash \neg\phi & \text{iff} & v \nvDash \phi \\
v \vDash \phi \wedge \psi & \text{iff} & v \vDash \phi \text{ and } v \vDash \psi \\
v \vDash \phi \vee \psi & \text{iff} & v \vDash \phi \text{ or } v \vDash \psi \\
v \vDash \phi \Rightarrow \psi & \text{iff} & v \vDash \phi \text{ implies } v \vDash \psi \\
v \vDash \phi \Leftrightarrow \psi & \text{iff} & v \vDash \phi \text{ iff } v \vDash \psi
\end{array}
$$

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

$$\vDash \phi \quad \text{iff} \quad v \vDash \phi \text{ for all } v.$$

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 $\llbracket \phi \rrbracket^v$. For example, under the assignment $v(\mathtt{p}_1) = 1, v(\mathtt{p}_2) = 0, v(\mathtt{p}_3) = 1$ we can calculate $\llbracket \phi \rrbracket^v$ 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 $\llbracket \phi \rrbracket^v = 1$.

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* ψ *any formula, then*

$$\Phi \vdash \psi \iff \Phi \vDash \psi.$$

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 $\Phi \vdash \psi$, meaning there is a finite derivation of ψ, all of the hypotheses of which are in the set Φ. Take a valuation v such that $v \vDash \Phi$, meaning that $v \vDash \phi$ for all $\phi \in \Phi$. Observe that for each rule of inference, for any valuation v, if $v \vDash \vartheta$ for all the hypotheses of the rule, then $v \vDash \gamma$ for the conclusion. By induction on the derivations therefore $v \vDash \phi$.

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

The key lemma is this:

Lemma 2.2.2 (Model Existence)**.** *A set* Φ *of formulas is* consistent*,* $\Phi \nvdash \bot$*, just if 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 that for every formula ψ, either $\psi \in \Psi$ or $\neg\psi \in \Psi$ and not both. Enumerate the formulas $\phi_0, \phi_1, ...,$ and let,

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

Now for each propositional variable $\mathtt{p}$, define $v(\mathtt{p}) = 1$ just if $\mathtt{p} \in \Psi$. $\square$

2.3 Boolean algebra

There is of course another apporach to propositional logic, which also goes back to the 19th century, namely that of Boolean algebra, which draws on the analogy between the propositional operations and the arithmetical ones.

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:

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

This is of course an algebraic theory, like those considered in the previous chapter. Familiar examples of Boolean algebras are $2 = \{0, 1\}$, with the usual operations, and more generally, any powerset $\mathcal{P}X$, with the set-theoretic operations $A \vee B = A \cup B$, etc. (indeed, $2 = \mathcal{P}1$ is a special case.).

Exercise 2.3.2. Show that the free Boolean algebra $B(n)$ on n-many generators is the double powerset $\mathcal{PP}(n)$, and determine the free functor on finite sets.

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 equational reasoning (Section **??**), from 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*,

$$\Phi \vdash^{=} \psi \tag{2.2}$$

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 connectives $\Rightarrow$ or $\Leftrightarrow$, let us replace them with their Boolean equivalents, by adding the equations:

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

For example, here is an equational proof of $(\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)$.

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

Thus,

$$\vdash^{=} (\phi \Rightarrow \psi) \vee (\psi \Rightarrow \phi)\,.$$

We now ask: *What is the relationship between equational provability $\Phi \vdash^{=} \phi$, deductive entailment $\Phi \vdash \phi$, and semantic entailment $\Phi \vDash \phi$?*

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,

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

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

2. Both

   $$\vdash^{=} \phi \Leftrightarrow \phi^{\vee} \qquad \text{and} \qquad \vdash^{=} \phi \Leftrightarrow \phi^{\wedge}\,.$$

Exercise 2.3.4. Using Exercise 2.3.3, show that for every propositional formula ϕ, equational provability is equivalent to semantic validity,

$$\vdash^{=} \phi \iff \vDash \phi\,.$$

Hint: Put ϕ into conjunctive normal form and read off a truth valuation that falsifies it, if there is one.

Exercise 2.3.5. 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, and that a finitely complete and cocomplete cartesian closed poset is a Boolean algebra just if it satisfies $x = (x \Rightarrow 0) \Rightarrow 0$, where, as before, we define $x \Rightarrow y := \neg x \vee y$. Finally, show that homomorphisms of Boolean algebras $f : B \to B'$ are the same thing as functors (i.e. monotone maps) that preserve all finite limits and colimits.

2.4 Lawvere duality for Boolean algebras

Let us apply the machinery of algebraic theories from Chapter **??** 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 $1, B, B^2, \ldots$, containing a Boolean algebra $\mathcal{B}$, with underlying object $|\mathcal{B}| = B$. By Theorem **??**, $\mathbb{B}$ has the universal property that finite product preserving (FP) functors from $\mathbb{B}$ into any FP-category $\mathbb{C}$ correspond (pseudo-)naturally to Boolean algebras in $\mathbb{C}$,

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

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

$$\frac{F : \mathbb{B} \longrightarrow \mathbb{C} \qquad \mathsf{FP}}{\overline{F(\mathcal{B}) \qquad\qquad \mathsf{BA}(\mathbb{C})}}$$

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

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

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

$$\mathcal{A}^\sharp(\mathcal{B}) \cong \mathcal{A}, \qquad F(\mathcal{B})^\sharp \cong F\,.$$

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

By Lawvere duality, Corollary **??**, 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 $F(n)$. 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 **??**).

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

$$\vdash^{=} \phi \iff \mathbb{B} \vdash \phi = 1 \iff \mathcal{B} \vDash \phi = 1\,.$$

As we showed in Proposition **??**, the image of the universal model $\mathcal{B}$ 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 $|\mathsf{y}_{\mathbb{B}}(\mathcal{B})| = \mathsf{Hom}_{\mathbb{B}}(-, B)$. By Proposition **??** 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 extend the above reasoning as follows:

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

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

$$\begin{aligned} |\mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(\mathcal{B})| &\cong \mathsf{eval}_{B^n}|\mathsf{y}_{\mathbb{B}}(\mathcal{B})| \cong \mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(|\mathcal{B}|) \cong \mathsf{eval}_{B^n}\mathsf{y}_{\mathbb{B}}(B) \cong \mathsf{y}_{\mathbb{B}}(B)(B^n) \\ &\cong \mathsf{Hom}_{\mathbb{B}}(B^n, B) \cong \mathsf{Hom}_{\mathsf{BA}^{\mathsf{op}}}(F(n), F(1)) \cong \mathsf{Hom}_{\mathsf{BA}}(F(1), F(n)) \cong |F(n)|\,. \end{aligned}$$

Thus to test for equational provability it suffices to check the equations in the free algebras $F(n)$ (which makes sense, since these are 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* $\vdash_{=} \phi$ *just in case, for every free Boolean algebra* $F(n)$*, we have* $F(n) \vDash \phi = 1$*.*

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 $F(n)$

describes the unique Boolean homomorphism

$$F(1) \xrightarrow{\overline{\phi}} F(k) \xrightarrow{\overline{(f_1, ..., f_k)}} F(n)\,,$$

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, we then have:

Proposition 2.4.2 (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,*

$$\vdash^{=} \phi\,,$$

just if it holds generally in any Boolean algebra (in Set*).*

Proof. By "holding generally" is meant the universal quantification of the equation over elements of a given Boolean algebra B, which is of course equivalent to saying that it holds for all elements of B, in the sense stated after the Lemma. But, as above, this is equivalent to the condition that for all $b_1, ..., b_k \in B$, for $\overline{(b_1, ..., b_k)} : F(k) \to B$ 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 $F(k)$. □

The analogous statement for equational entailment $\Phi \vdash_{=} \phi$ is left as an exercise.

Corollary 2.4.2 is a (very) special case of the Gödel completeness theorem for first-order logic, for *just* the equational fragment of *just* the specific theory of Boolean algebras (although, an analogous result of course holds for any other algebraic theory, and many other systems of logic can be reduced to the algebraic case). Nonetheless, it suggests another approach to the semantics of propositional logic based upon the idea of a *Boolean valuation*, generalizing the traditional truth-value semantics from Section 2.2. We pursue this idea systematically in the following section.

Exercise 2.4.3. For a formula in context $\mathtt{p}_1, ..., \mathtt{p}_k \mid \vartheta$ and a Boolean algebra $\mathcal{A}$, let the expression $\vartheta[a_1/\mathtt{p}_1, ..., a_k/\mathtt{p}_k]$ denote the element of $\mathcal{A}$ resulting from interpreting the propositional variables $\mathtt{p}_i$ in the context as the elements a_i of $\mathcal{A}$, and evaluating the resulting expression using the Boolean operations of $\mathcal{A}$. 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 $\Phi \vdash^{=} \psi$ means that $\psi = 1$ is equationally provable from the set of equations $\{\phi = 1 \mid \phi \in \Phi\}$. Show that $\Phi \vdash^{=} \psi$ 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 $\Phi \vdash^{=} \psi$ for not necessarily finite sets Φ.

2.5 Functorial semantics for propositional logic

Considering the algebraic theory of Boolean algebras suggests 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 finite product categories, namely that represented categorically by *poset* categories with finite products $\wedge$ *and coproducts* $\vee$ (each of these specializations could, of course, also be considered separately, giving $\wedge$-semi-lattices and categories with finite products $\times$ and coproducts $+$, respectively). Thus we are moving from the top right corner to the bottom center position in the following Hasse diagram of structured categories:

$$
\begin{tikzcd}
+ \arrow[dr, dash] \arrow[dd, dash] & & \times \arrow[dl, dash] \arrow[dd, dash] \\
& +\,,\times \arrow[dd, dash] & \\
\vee \arrow[dr, dash] & & \wedge \arrow[dl, dash] \\
& \vee\,,\wedge &
\end{tikzcd}
$$

In Chapter **??** we shall see how first-order logic results categorically from these two cases by "indexing the lower one over the upper one".

Definition 2.5.1. 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 *consequences* $\Phi \vdash_{\mathbb{T}} \phi$ are those judgements that are derivable by natural deduction (as in Section 2.1), from the axioms $A_{\mathbb{T}}$.

Definition 2.5.2. 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:

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

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 2.5.3. *The functor* $\mathsf{Mod}(\mathbb{T}) : \mathsf{BA} \to \mathsf{Set}$ *is representable, with representing Boolean algebra* $\mathcal{B}_{\mathbb{T}}$*, called the* Lindenbaum-Tarski algebra *of* $\mathbb{T}$*.*

Proof. We construct $\mathcal{B}_{\mathbb{T}}$ in two steps:

Step 1: Suppose first that $A_{\mathbb{T}}$ is empty, so $\mathbb{T}$ is just a set V of propositional variables. Define the *Lindenbaum-Tarski 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*,

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

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*,

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

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

Now observe that the construction of $\mathcal{B}_{\mathbb{T}}$ is a variation on that of the *syntactic category* $\mathcal{C}_{\mathbb{T}}$ of the algebraic theory $\mathbb{T}$ in the sense of the previous chapter, and the statement of the theorem is its universal property as the classifying category of $\mathbb{T}$-models, namely

$$\mathsf{Mod}(\mathbb{T}, \mathcal{B}) \cong \mathsf{Hom}_{\mathsf{BA}}(\mathcal{B}_{\mathbb{T}}, \mathcal{B})\,, \tag{2.8}$$

naturally in $\mathcal{B}$. (Indeed, 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 **??** from Chapter 1. Further details are given in the following Remark 2.5.4 for the interested reader. □

Remark 2.5.4 (Adjoint Rules for Propositional Calculus)**.** For the construction of the Lindenbaum-Tarski 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 hypotheses 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).

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

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

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*:

$$\frac{}{\neg\neg\phi \vdash \phi} \tag{2.9}$$

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 2.5.5. Fill in the details of the proof that $\mathcal{B}_{\mathbb{T}}$ is a well-defined Boolean algebra, with the universal property stated in (2.8).

Just as for the case of algebraic theories and FP categories, we now have the following corollary of the classifying theorem 2.5.3. (Note that the recipe at (2.6) for a Boolean valuation in $F(n)$ of the formula in context $\mathtt{p}_1, ..., \mathtt{p}_k \mid \phi$ is exactly a *model* in $F(n)$ of the theory $\mathbb{T} = \{\mathtt{p}_1, ..., \mathtt{p}_k\}$.)

Corollary 2.5.6. *For any set of formulas Φ and formula ϕ, derivability $\Phi \vdash \phi$ is equivalent to validity under all Boolean valuations. Therefore by Proposition 2.4.2 (and Exercise 2.4.3), we also have*

$$\Phi \vdash \phi \iff \Phi \vdash^{=} \phi\,.$$

Remark 2.5.7. If $A_{\mathbb{T}}$ is non-empty, but finite, then let

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

We then have

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

where as usual $\mathcal{B}/b$ denotes the slice category of the Boolean algebra $\mathcal{B}$ over an element $b \in \mathcal{B}$.

Remark 2.5.8. Our definition of the Lindenbaum-Tarski algebra is given in terms of *provability*, rather than the more familiar semantic definition using (truth) valuations. The two are, of course, equivalent in light of Theorem 2.2.1, but since we intend to prove that theorem, this definition will be more useful, as it parallels that of the syntactic category $\mathcal{C}_{\mathbb{T}}$ of an algebraic theory.

Inspecting the universal property (2.8) of $\mathcal{B}_{\mathbb{T}}$ for the case $\mathcal{B}[V]$ where there are no axioms, we now have the following.

Corollary 2.5.9. *The Lindenbaum-Tarski algebra $\mathcal{B}[V]$ is the free Boolean algebra on the set V. In particular, $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n]$ is the finitely generated, free Boolean algebra $F(n)$.*

The isomorphism $\mathcal{B}[\mathtt{p}_1, ..., \mathtt{p}_n] \cong F(n)$ expresses the fact recorded in Corollary 2.5.6 that the relations of derivability by natural deduction $\Phi \vdash \phi$ and equational provability $\Phi \vdash^{=} \phi$ agree — answering part of the question at the end of Section **??**.

Exercise 2.5.10. Show that the 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.

Finally, we can use the following to finish the comparison of $\vdash \phi$ and $\vDash \phi$.

Lemma 2.5.11. *Let $\mathcal{B}$ be a* finitely presented *Boolean algebra in which $0 \neq 1$. Then there is a Boolean homomorphism*

$$h : \mathcal{B} \to \mathbb{2} \,.$$

Proof. By Exercise 2.5.10, we can assume that $\mathcal{B} = \mathcal{B}[\mathtt{p}_1 ... \mathtt{p}_n]/\alpha$ classifying the theory $\mathbb{T} = (\mathtt{p}_1 ... \mathtt{p}_n, \alpha)$. By the assumption that $0 \neq 1$ in $\mathbb{B} = \mathcal{B}[\mathtt{p}_1 ... \mathtt{p}_n]/\alpha$, we have $\alpha \neq 0$ in the free Boolean algebra $F(n) \cong \mathcal{B}[\mathtt{p}_1 ... \mathtt{p}_n]$, whence $\alpha \nvdash \bot$. Since $F(n) \cong \mathcal{P}\mathcal{P}(n)$, there is a valuation $\vartheta : \{\mathtt{p}_1 ... \mathtt{p}_n\} \to \mathbb{2}$ such that $[\![\alpha]\!]^{\vartheta} = 1$. This is exactly a Boolean homomorphism $\mathcal{B}[\mathtt{p}_1 ... \mathtt{p}_n]/\alpha \to \mathbb{2}$, as required. □

Corollary 2.5.12. *For any set of formulas Φ and formula ϕ, derivability $\Phi \vdash \phi$ is equivalent to semantic entailment,*

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

Proof. By 2.5.6, it suffices to show that $\Phi \vDash \phi$ is equivalent to $\Phi \vdash^{=} \phi$, but the latter we know to be equivalent to holding in all Boolean valuations in free Boolean algebras $F(n)$, and the former to holding in all *truth* valuations, i.e. Boolean valuations in $\mathbb{2}$. Thus it will suffice to embed $F(n)$ as a Boolean algebra into a powerset $\mathcal{P}X = \mathbb{2}^X$, for a set X. By Lemma 2.5.11 we can take $X = 2^n$. □

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 $\mathbb{2}$-valued, we have a factorization,

$$\mathcal{B} \hookrightarrow \mathbb{2}^{\mathcal{B}^{\mathsf{op}}} \hookrightarrow \mathsf{Set}^{\mathcal{B}^{\mathsf{op}}}$$

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 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.

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

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 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 $\mathbb{2}^{\mathcal{B}^{\mathsf{op}}} = \mathsf{Pos}(\mathcal{B}^{\mathsf{op}}, \mathbb{2})$ of all *contravariant*, monotone maps $\mathcal{B}^{\mathsf{op}} \to \mathbb{2}$ (which indeed is $\mathsf{Sub}(1) \hookrightarrow \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$) is easily seen to be isomorphic to the poset $\downarrow\!\mathcal{B}$ of all *sieves* (or "downsets") 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

$$\mathsf{Pos}(\mathcal{B}^{\mathsf{op}}, \mathbb{2}) \cong \downarrow\!\mathcal{B} \tag{2.10}$$

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 (2.10) is indeed an isomorphism of posets, and that it takes 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 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 $\{1\} \hookrightarrow \mathbb{2}$, such homomorphisms correspond to *filters* in $\mathcal{B}$: "upsets" that are also closed under $\wedge$. The representables then correspond to the *principal filters* $\uparrow b \subseteq \mathcal{B}$. The problem with using this approach for a completeness theorem for *propositional* logic 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* Bool *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 $b \wedge \neg b = 0$. $\square$

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 $b \vee b' \in F$ 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.

If $\mathcal{B}$ is Boolean, it then follows 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 $(f : \mathcal{B} \to \mathbb{2}) \mapsto F_f := f^{-1}(1) \subseteq \mathcal{B}$. The prime filter F_f may be called the *(filter) kernel* of $f : \mathcal{B} \to \mathbb{2}$.

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

1. *F is a prime filter*
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. *F is a filter, and for each $b \in \mathcal{B}$, either $b \in F$ or $\neg b \in \mathcal{F}$ and not both.*
5. *F is a* maximal filter*: F is a filter and for all filters G, if $F \subseteq G$ 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 (2.10)) 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:

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

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 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 $\mathcal{B} \rightarrowtail \mathcal{P}X$ into a powerset.*

Proof. Let X be the set of prime filters in $\mathcal{B}$ and consider the map $h : \mathcal{B} \to \mathcal{P}X$ given by $h(b) = \{F \mid b \in F\}$. 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 $\mathcal{B} \rightarrowtail \mathcal{P}(X_{\mathcal{B}})$ the powerset Boolean algebra

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

is evidently (covariantly) functorial in $\mathcal{B}$, and has an apparent "double-dual" form $\mathcal{B}^{**}$. This suggests a possible duality between the categories Bool and Set,

$$\begin{tikzcd} \mathsf{Bool}^{\mathsf{op}} \arrow[r, bend left, "*"] & \mathsf{Set} \arrow[l, bend left, "*"] \end{tikzcd} \tag{2.11}$$

with contravariant functors $\mathcal{B}^* = \mathsf{Bool}(\mathcal{B}, \mathbb{2})$, the set of prime filters, for a Boolean algebra $\mathcal{B}$, and $S^* = \mathsf{Set}(S, 2)$, the powerset Boolean algebra, for a set S. This indeed gives a contravariant adjunction "on the right",

$$\begin{array}{cc} \mathcal{B} \to \mathcal{P}S & \mathsf{Bool} \\ \hline\hline S \to X_{\mathcal{B}} & \mathsf{Set} \end{array} \tag{2.12}$$

by applying the contravariant functors

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

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

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

The homomorphism $\eta_{\mathcal{B}}$ takes an element $b \in \mathcal{B}$ to the set of prime filters that contain it, and the function ε_S takes an element $s \in S$ to 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 (2.14).

The adjunction (2.14) is not an equivalence, however, because neither of the units $\eta_{\mathcal{B}}$ nor ε_S is in general an isomorphism. We can do better by topologizing the set $X_{\mathcal{B}}$ of prime filters, in order to be able to cut down the powerset $\mathcal{P}X_{\mathcal{B}} \cong \mathsf{Set}(X_{\mathcal{B}}, 2)$ to just the *continuous* functions into the discrete space 2, which then correspond to the clopen sets in $X_{\mathcal{B}}$. To do so, we take as *basic open sets* all those sets of the form:

$$B_b = \{P \in X_{\mathcal{B}} \mid b \in P\}, \qquad b \in \mathcal{B}. \tag{2.13}$$

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.

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 (2.13), 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, with the principal ideals $\downarrow b$ corresponding exactly to the clopen sets.*

Proof. Exercise! □

We now have an improved adjunction

$$\begin{tikzcd} \mathsf{Bool}^{\mathsf{op}} \arrow[r, bend left, "\mathsf{Spec}"] & \mathsf{Top} \arrow[l, bend left, "\mathsf{Clop}"] \end{tikzcd} \tag{2.14}$$

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

for which, up to isomorphism, the space $\mathsf{Spec}(\mathcal{B})$ has the underlying set $\mathsf{Bool}(\mathcal{B}, \mathbb{2})$ 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 topological 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 (and others to be met later on), the role of dualizing object is played by the category Set of all sets.

Toward the goal of determining the image of the functor $\mathsf{Spec} : \mathsf{Bool}^{\mathsf{op}} \to \mathsf{Top}$, 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,

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

and is therefore a compact Hausdorff space by Tychonoff's theorem. Indeed, the basis (2.13) 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.

In order to further cut down the adjunction on the topological side, we can now restrict it 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* Bool *and* Stone*,*

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

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

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

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

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

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 (2.13), 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 $\mathcal{B}'$.

Let S be a Stone space and consider the continuous function

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

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 $S - K$ is therefore closed, and thus compact, and so is its image $\varepsilon_S(S - K)$, which is therefore closed. But since ε_S is a bijection, $\varepsilon_S(S - K)$ is the complement of $\varepsilon_S(K)$, which is therefore open. $\square$

Remark 2.7.7. Another way to cut down the adjunction (2.14),

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

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

$$\mathsf{CABool}^{\mathsf{op}} \simeq \mathsf{Set},$$

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

Remark 2.7.8. See Johnstone [**?**] for a more detailed presentation of the material in this section (and much more). Also see [**?**] 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 [**?**]

Chapter 3

λ-Calculus

3.1 Categorification and the Curry-Howard correspondence

Consider the following natural deduction proof in propositional calculus.

$$\dfrac{\dfrac{\dfrac{\dfrac{[(A \wedge B) \wedge (A \Rightarrow B)]^1}{A \wedge B}}{A} \qquad \dfrac{[(A \wedge B) \wedge (A \Rightarrow B)]^1}{A \Rightarrow B}}{B}}{(A \wedge B) \wedge (A \Rightarrow B) \Rightarrow B}\,(1)$$

This deduction shows that

$$\vdash (A \wedge B) \wedge (A \Rightarrow B) \Rightarrow B.$$

But so does the following:

$$\dfrac{\dfrac{\dfrac{[(A \wedge B) \wedge (A \Rightarrow B)]^1}{A \Rightarrow B} \qquad \dfrac{\dfrac{[(A \wedge B) \wedge (A \Rightarrow B)]^1}{A \wedge B}}{A}}{B}}{(A \wedge B) \wedge (A \Rightarrow B) \Rightarrow B}\,(1)$$

As does:

$$\dfrac{\dfrac{\dfrac{[(A \wedge B) \wedge (A \Rightarrow B)]^1}{A \wedge B}}{B}}{(A \wedge B) \wedge (A \Rightarrow B) \Rightarrow B}\,(1)$$

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 $\vdash \phi$ 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 is effected by annotating the proofs with *proof-terms* as they are constructed, as follows:

$$\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_2(x):A\Rightarrow B} \qquad \dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_1(\pi_1(x)):A}}{\pi_2(x)(\pi_1(\pi_1(x))):B}}{\lambda x.\pi_2(x)(\pi_1(\pi_1(x))):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\ (1)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_1(\pi_1(x)):A} \qquad \dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_2(x):A\Rightarrow B}}{\pi_2(x)(\pi_1(\pi_1(x))):B}}{\lambda x.\pi_2(x)(\pi_1(\pi_1(x))):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\ (1)$$

$$\dfrac{\dfrac{\dfrac{[x:(A\wedge B)\wedge(A\Rightarrow B)]^1}{\pi_1(x):A\wedge B}}{\pi_2(\pi_1(x)):B}}{\lambda x.\pi_2(\pi_1(x)):(A\wedge B)\wedge(A\Rightarrow B)\Rightarrow B}\ (1)$$

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 $\phi \vdash \psi$ is a mere preorder. The addition of proof terms $x:\phi \vdash t:\psi$ results in a *categorification* of this preorder, in the sense that it is a "proper" category, the preordered reflection of which is the deducibility preorder. And now the following remarkable fact emerges: it is hardly surprising that the deducibility preorder has, say, finite products $\phi\wedge\psi$ or even exponentials $\phi\Rightarrow\psi$; but it is *amazing* that the category with proof terms $x:\phi\vdash t:\psi$ 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. The calculus of proof terms can be presented formally in a system of *simple type theory*, with an alternate interpretation as a formal system of function application and abstraction. This dual interpretation—as the proof theory of propositional logic, and as a system of type theory for the specification of functions—is called the *Curry-Howard correspondence* []. From the categorical point of view, it expresses the structural equivalence between the cartesian closed categories of proofs in propositional logic and terms in simple type theory. Both of these can be seen as categorifications of their preorder reflection, the deducibility preorder of propositional logic (cf. [**?**]).

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.

3.2 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:

$$f \leq g \iff fx \leq gx \quad \text{for all } x \in P\,. \qquad (f, g : P \to Q)$$

Thus $\mathsf{Hom}(P, Q)$ is again an object of Pos, when equipped with a suitable order.

Similarly, given monoids $K, M \in \mathsf{Mon}$, there is a natural monoid structure on the set $\mathsf{Hom}(K, M)$, defined pointwise by

$$(f \cdot g)x = fx \cdot gx\,. \qquad (f, g : K \to M,\, x \in K)$$

Thus the category Mon also admits such "internal Hom"s. 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 "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 $\epsilon : B^A \times A \to B$ which evaluates a morphism $f \in B^A$ at an argument $x \in A$ to give a value $\epsilon \circ \langle f, x\rangle \in B$. This is always going to be present for the underlying functions if we're starting from a set of functions $\mathsf{Hom}(A, B)$, but it 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 3.2.1. 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 $f = \epsilon \circ (\widetilde{f} \times 1_A)$.

Definition 3.2.1 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

$$f\langle z, x\rangle = (\widetilde{f}z)x \; .$$

That is all there is to it, really, except that variables and elements never need to be mentioned. The benefit of this is that the definition is applicable 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 3.2.2. Suppose $\mathcal{C}$ has all finite products. An object $A \in \mathcal{C}$ is *exponentiable* when the exponential B^A exists for every $B \in \mathcal{C}$. 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 3.2.3. 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 $\widetilde{h} = u : 1 \to \mathsf{Hom}(M, N)$. So Mon cannot be cartesian closed. The same argument works in the category Group, and in many related ones. (But see **??** below on one way of embedding Group into a CCC.)

Exercise 3.2.4. Is the evaluation function $\mathsf{eval} : \mathsf{Hom}(M, N) \times M \to N$ a homomorphism of monoids?

[1] Also, f is called the transpose of $\widetilde{f}$, so that f and $\widetilde{f}$ are each other's transpose.

Two characterizations of CCCs

Proposition 3.2.5. *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 $\epsilon : -^A \times A \Longrightarrow \mathsf{1}_{\mathcal{C}}$ is the counit of the adjunction. The universal property of the exponential is precisely the universal property of the counit ϵ.

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.

$$\begin{tikzcd}
B^A \times A \arrow[r, "\epsilon_B"] \arrow[d, "g^A \times \mathsf{1}_A"'] & B \arrow[d, "g"] \\
C^A \times A \arrow[r, "\epsilon_C"'] & C
\end{tikzcd} \tag{3.1}$$

The counit $\epsilon : -^A \times A \Longrightarrow \mathsf{1}_{\mathcal{C}}$ at B is then ϵ_B itself, and the naturality square for ϵ is then exactly (3.1), i.e. the defining property of $(f \circ \epsilon_B)^\sim$:

$$\epsilon_C \circ (g^A \times \mathsf{1}_A) = \epsilon_C \circ ((g \circ \epsilon_B)^\sim \times \mathsf{1}_A) = g \circ \epsilon_B \ .$$

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 3.2.6. *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 "equational" in a sense that can be made precise; see [**?**]). We can therefore give an explicit, equational formulation of cartesian closed categories.

Proposition 3.2.7 (Equational version of CCC)**.** *A category $\mathcal{C}$ is cartesian closed if, and only if, it has the following structure:*

1. *An object $1 \in \mathcal{C}$ and a morphism $!_A : A \to 1$ 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 $f : A \to 1$,*
$$f = !_A \, .$$
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

$$\pi_0 \circ \langle f, g \rangle = f \qquad \textit{and} \qquad \pi_1 \circ \langle f, g \rangle = g$$

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 3.2.8. Use the equational characterization of CCCs, Proposition 3.2.7, 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?

3.3 Positive propositional calculus

We begin with the example of a cartesian closed poset and a first application to propositional logic.

Example 3.3.1. Consider the *positive propositional calculus* PPC with conjunction and implication, as in Section 2.1. Recall that PPC is the set of all propositional formulas ϕ constructed from propositional variables $p_1, p_2, \ldots$, 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 for instance by the natural deduction system **??** of section **??**. 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 just the usual *Lindenbaum-Tarski* algebra of the system of 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 we have $\phi \wedge \psi \vdash \phi$ 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.

We have already remarked that implication is right adjoint to conjunction in propositional calculus,

$$(-) \wedge \phi \ \dashv \ \phi \Rightarrow (-) \ . \tag{3.2}$$

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 have now shown:

Proposition 3.3.2. *The poset $\mathcal{C}_{\mathsf{PPC}}$ of positive propositional calculus is cartesian closed.*

Let us now use this fact to show that the positive propositional calculus is *deductively complete* with respect to the following notion of *Kripke semantics* [].

Definition 3.3.3 (Kripke model)**.** Let K be a poset. Suppose we have a relation

$$k \Vdash p$$

between elements $k \in K$ and propositional variables p, such that

$$j \leq k,\ k \Vdash p \quad \text{implies} \quad j \Vdash p. \tag{3.3}$$

Extend $\Vdash$ to all formulas ϕ in PPC by defining

$$\begin{array}{lll} k \Vdash \top & \text{always}, & \\ k \Vdash \phi \wedge \psi & \text{iff} & k \Vdash \phi \text{ and } k \Vdash \psi\,, \\ k \Vdash \phi \Rightarrow \psi & \text{iff} & \text{for all } j \leq k, \text{ if } j \Vdash \phi, \text{ then } j \Vdash \psi\,. \end{array} \tag{3.4}$$

Finally, say that ϕ *holds on* K, written

$$K \Vdash \phi$$

if $k \Vdash \phi$ for all $k \in K$ (for all such relations $\Vdash$).

Theorem 3.3.4 (Kripke completeness for PPC)**.** *A propositional formulas ϕ is provable from the rules of deduction for PPC if, and only if, $K \Vdash \phi$ for all posets K. Briefly:*

$$\mathsf{PPC} \vdash \phi \quad \textit{iff} \quad K \Vdash \phi \textit{ for all } K.$$

We will require the following (which extends the discussion in Section 2.6).

Lemma 3.3.5. *For any poset P, the poset $\downarrow P$ of all downsets in P, ordered by inclusion, is cartesian closed. Moreover, the downset embedding,*

$$\downarrow(-) : P \to \downarrow P$$

preserves any CCC structure that exists in P.

Proof. The total downset P is obviously terminal, and for any downsets $S, T \in \downarrow P$, the intersection $S \cap T$ is also closed down, so we have the products $S \wedge T = S \cap T$. For the exponential, set

$$S \Rightarrow T = \{p \in P \mid \downarrow(p) \cap S \subseteq T\}.$$

Then for any downset Q we have

$$Q \subseteq S \Rightarrow T \quad \text{iff} \quad \downarrow(q) \cap S \subseteq T, \text{ for all } q \in Q. \tag{3.5}$$

But that means that

$$\textstyle\bigcup_{q \in Q}(\downarrow(q) \cap S) \subseteq T\,,$$

which is equivalent to $Q \cap S \subseteq T$, since $\bigcup_{q \in Q}(\downarrow(q) \cap S) = (\bigcup_{q \in Q} \downarrow(q)) \cap S = Q \cap S$.

The preservation of CCC structure by $\downarrow(-) : P \to \downarrow P$ follows from its preservation by the Yoneda embedding, of which $\downarrow(-)$ is a factor,

$$\begin{tikzcd} & \mathsf{Set}^{P^{\mathsf{op}}} \\ P \arrow[ur, hook, "\mathsf{y}"] \arrow[r, "\downarrow(-)"'] & \downarrow P \arrow[u, hook] \end{tikzcd}$$

But it is also easy enough to check directly: preservation of any limits 1, $p \wedge q$ that exist in P are clear. Suppose $p \Rightarrow q$ is an exponential; then for any downset D we have:

$$\begin{array}{lll} D \subseteq \downarrow(p \Rightarrow q) & \text{iff} & \downarrow(d) \subseteq \downarrow(p \Rightarrow q)\ , \text{ for all } d \in D \\ & \text{iff} & d \leq p \Rightarrow q\ , \text{ for all } d \in D \\ & \text{iff} & d \wedge p \leq q\ , \text{ for all } d \in D \\ & \text{iff} & \downarrow(d \wedge p) \subseteq \downarrow(q)\ , \text{ for all } d \in D \\ & \text{iff} & \downarrow(d) \cap \downarrow(p) \subseteq \downarrow(q)\ , \text{ for all } d \in D \\ & \text{iff} & D \subseteq \downarrow(p) \Rightarrow \downarrow(q) \end{array}$$

where the last line is by (3.5). (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 3.3.4) 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 $\mathcal{C}_{\mathsf{PPC}} \to \downarrow K$, which by Step 1 is just an arbitrary map $\mathsf{Var} \to \downarrow K$, as in (3.3). 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,
$$\frac{[\![-]\!] : \mathcal{C}_{\mathsf{PPC}} \longrightarrow \downarrow K \cong 2^{K^{\mathsf{op}}}}{\Vdash : K^{\mathsf{op}} \times \mathcal{C}_{\mathsf{PPC}} \longrightarrow 2}$$
where we use the poset 2 to classify downsets in a poset P (via upsets in P^{op}),
$$\downarrow P \cong 2^{P^{\mathsf{op}}} \cong \mathsf{Pos}(P^{\mathsf{op}}, 2)\,,$$
by taking the 1-kernel $f^{-1}(1) \subseteq P$ of a monotone map $f : P^{\mathsf{op}} \to 2$. (The contravariance will be convenient in Step 3). Note that the monotonicity of $\Vdash$ yields the conditions
$$p \leq q\,,\ q \Vdash \phi \implies p \Vdash \phi$$

and

$$p \Vdash \phi\,,\ \phi \vdash \psi \implies p \Vdash \psi\,.$$

and the CCC preservation of the transpose $[\![-]\!]$ yields the Kripke forcing conditions (3.4) (exercise!).

3. For any model $(K, \Vdash)$, by the adjunction in (2) we then have $K \Vdash \phi$ iff $[\![\phi]\!] = K$, the total downset.

4. Because the downset/Yoneda embedding $\downarrow$ preserves the CCC structure (by Lemma 3.3.5), $\mathcal{C}_{\mathsf{PPC}}$ has a *canonical model*, namely $(\mathcal{C}_{\mathsf{PPC}}, \Vdash)$, where:

$$\frac{\downarrow(-)\ :\ \mathcal{C}_{\mathsf{PPC}} \longrightarrow \downarrow\mathcal{C}_{\mathsf{PPC}} \cong 2^{\mathcal{C}_{\mathsf{PPC}}{}^{\mathsf{op}}} \hookrightarrow \mathsf{Set}^{\mathcal{C}_{\mathsf{PPC}}{}^{\mathsf{op}}}}{\Vdash\ :\ \mathcal{C}_{\mathsf{PPC}}{}^{\mathsf{op}} \times \mathcal{C}_{\mathsf{PPC}} \longrightarrow 2 \hookrightarrow \mathsf{Set}}$$

5. Now note that for the Kripke relation $\Vdash$ in (4), we have $\Vdash\, = \,\vdash$, since it's essentially the transpose of the Yoneda embedding. Thus the model is logically generic, in the sense that $\mathcal{C}_{\mathsf{PPC}} \Vdash \phi$ iff $\mathsf{PPC} \vdash \phi$.

□

Exercise 3.3.6. Verify the claim that CCC preservation of the transpose $[\![-]\!]$ of $\Vdash$ yields the Kripke forcing conditions (3.4).

Exercise 3.3.7. Give a countermodel to show that $\mathsf{PPC} \nvdash (\phi \Rightarrow \psi) \Rightarrow \phi$

3.4 Heyting algebras

We now extend the positive propositional calculus to the full intuitionistic propositional calculus. This involves adding the finite coproducts 0 and $p \vee q$ to 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 3.3.5, 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 meet $\wedge$ and join $\vee$, satisfying for all $x, y, z \in L$,

$$0 \leq x \leq 1 \qquad \frac{z \leq x \qquad z \leq y}{\overline{\overline{z \leq x \wedge y}}} \qquad \frac{x \leq z \qquad x \leq y}{\overline{\overline{x \vee y \leq z}}}$$

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.

A lattice can be axiomatized equationally as a set with two distinguished elements 0 and 1 and two binary operations $\wedge$ and $\vee$, satisfying the following equations:

$$\begin{aligned}
(x \wedge y) \wedge z &= x \wedge (y \wedge z) , & (x \vee y) \vee z &= x \vee (y \vee z) , \\
x \wedge y &= y \wedge x , & x \vee y &= y \vee x , \\
x \wedge x &= x , & x \vee x &= x , \\
1 \wedge x &= x , & 0 \vee x &= x , \\
& x \wedge (y \vee x) = x = (x \wedge y) \vee x .
\end{aligned} \tag{3.6}$$

The partial order on L is then determined by

$$x \leq y \iff x \wedge y = x .$$

Exercise 3.4.1. Show that in a lattice $x \leq y$ if, and only if, $x \wedge y = x$ if, and only if, $x \vee y = y$.

A lattice is *distributive* if the following distributive laws hold in it:

$$\begin{aligned}
(x \vee y) \wedge z &= (x \wedge z) \vee (y \wedge z) , \\
(x \wedge y) \vee z &= (x \vee z) \wedge (y \vee z) .
\end{aligned} \tag{3.7}$$

It turns out that if one distributive law holds then so does the other [**?**, I.1.5].

A *Heyting algebra* is a cartesian closed lattice H. This means that it has an operation $\Rightarrow$, satisfying for all $x, y, z \in H$

$$\frac{z \wedge x \leq y}{\overline{\overline{z \leq x \Rightarrow y}}}$$

A *Heyting algebra homomorphism* is a lattice homomorphism $f : K \to H$ 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.

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 (3.6), as well as the following ones for $\Rightarrow$.

$$\begin{aligned}(x \Rightarrow x) &= 1 \ ,\\ x \wedge (x \Rightarrow y) &= x \wedge y \ ,\\ y \wedge (x \Rightarrow y) &= y \ ,\\ (x \Rightarrow (y \wedge z)) &= (x \Rightarrow y) \wedge (x \Rightarrow z) \ .\end{aligned} \tag{3.8}$$

For a proof, see [**?**, I.1], where one can also find a proof that every Heyting algebra is distributive (exercise!).

Example 3.4.2. We know from Lemma 3.3.5 that for any poset P, the poset $\downarrow P$ of all downsets in P, ordered by inclusion, is cartesian closed. Moreover, we know that $\downarrow P \cong \mathbb{2}^{P^{\mathsf{op}}}$, as a poset, with the reverse pointwise ordering on monotone maps $P^{\mathsf{op}} \to \mathbb{2}$, or equivalently, $\downarrow P \cong \mathbb{2}^P$, with the functions ordered pointwise. Since $\mathbb{2}$ is a lattice, we can also take joins $f \vee g$ pointwise, in order to get joins in $\mathbb{2}^P$, which then correspond to finite unions of the corresponding downsets $f^{-1}\{0\} \cup g^{-1}\{0\}$. Thus, in sum, for any poset P, the lattice $\downarrow P \cong \mathbb{2}^P$ is a Heyting algebra, with the downsets ordered by inclusion, and the functions ordered pointwise.

Intuitionistic propositional calculus

There is a forgetful functor $U : \mathsf{Heyt} \to \mathsf{Set}$ which maps a Heyting algebra to its underlying set, and a homomorphism of Heyting algebras to the underlying function. Because Heyting algebras are models of an equational theory, there is a left adjoint $H \dashv U$, which is the usual "free" construction mapping a set S to the free Heyting algebra HS generated by it. As for all algebraic strictures, the construction of HS can be performed in two steps: first, define a set HS of formal expressions, and then quotient it by an equivalence relation generated by the axioms for Heyting algebras.

Thus let HS be the set of formal expressions generated inductively by the following rules:

1. Generators: if $x \in S$ then $x \in HS$.
2. Constants: $\bot, \top \in HS$.
3. Connectives: if $\phi, \psi \in HS$ then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \Rightarrow \psi) \in HS$.

We impose an equivalence relation on HS, which we write as equality $=$ and think of as such; it is defined as the smallest equivalence relation satisfying axioms (3.6) and (3.8). This forces HS to be a Heyting algebra. We define the action of the functor H on morphisms as usual: a function $f : S \to T$ is mapped to the Heyting algebra morphism $Hf : HS \to HT$ defined by

$$(Hf)\bot = \bot \ , \qquad (Hf)\bot = \bot \ , \qquad (Hf)x = fx \ ,$$
$$(Hf)(\phi \star \psi) = ((Hf)\phi) \star ((Hf)\psi) \ ,$$

where $\star$ stands for $\wedge$, $\vee$ or $\Rightarrow$.

The inclusion $\eta_S : S \to U(HS)$ of generators into the underlying set of the free Heyting algebra HS 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 K and an arbitrary function $f : S \to UK$. Then the Heyting algebra homomorphism $\overline{f} : HS \to K$ defined by

$$\overline{f}\bot = \bot\,, \qquad \overline{f}\bot = \bot\,, \qquad \overline{f}x = fx\,,$$
$$\overline{f}(\phi \star \psi) = (\overline{f}\phi) \star (\overline{f}\psi)\,,$$

where $\star$ stands for $\wedge$, $\vee$ or $\Rightarrow$, makes the following triangle commute:

$$\begin{tikzcd}
S \arrow[r, "\eta_S"] \arrow[dr, "f"'] & U(HS) \arrow[d, "U\overline{f}"] \\
& K
\end{tikzcd}$$

It is the unique such morphism because any two homomorphisms from HS which agree on generators must be equal. This is proved by induction on the structure of the formal expressions in HS.

We may now define the *intuitionistic propositional calculus* IPC to be the free Heyting algebra IPC on countably many generators p_0, p_1, …, called *atomic propositions* or *propositional variables*. This is a somewhat unorthodox definition from a logical point of view—normally we would start from a *calculus* consisting of a formal language, 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 describe IPC in the conventional way. The formulas of IPC are built inductively from propositional variables p_0, p_1, …, constants falsehood $\bot$ and truth $\top$, and binary operations conjunction $\wedge$, disjunction $\vee$ and implication $\Rightarrow$. The basic judgment of IPC is *logical entailment*

$$u_1 : A_1, \ldots, u_k : A_k \vdash B$$

which means "hypotheses A_1, …, A_k entail proposition B". The hypotheses are labeled with distinct labels u_1, …, u_k so that we can distinguish them, which is important when the same hypothesis appears more than once. Because the hypotheses are labeled it is irrelevant in what order they are listed, as long as the labels are not getting mixed up. Thus, the hypotheses $u_1 : A \vee B, u_2 : B$ are the same as the hypotheses $u_2 : B, u_1 : A \vee B$, but different from the hypotheses $u_1 : B, u_2 : A \vee B$. Sometimes we do not bother to label the hypotheses.

The left-hand side of a logical entailment is called the *context* and the right-hand side is the *conclusion*. Thus logical entailment is a relation between contexts and conclusions. The context may be empty. If Γ is a context, u is a label which does not occur in Γ, and A is a formula, then we write $\Gamma, u : A$ for the context Γ extended by the hypothesis $u : A$.

Definition 3.4.3. *Deductive entailment* is the smallest relation satisfying the following rules:

1. Conclusion from a hypothesis:
$$\frac{}{\Gamma \vdash A} \text{ if } u : A \text{ occurs in } \Gamma$$

2. Truth:
$$\frac{}{\Gamma \vdash \top}$$

3. Falsehood:
$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

4. Conjunction:
$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

5. Disjunction:
$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \qquad \frac{\Gamma \vdash A \vee B \qquad \Gamma, u : A \vdash C \qquad \Gamma, v : B \vdash C}{\Gamma \vdash C}$$

6. Implication:
$$\frac{\Gamma, u : A \vdash B}{\Gamma \vdash A \Rightarrow B} \qquad \frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

A *proof* of $\Gamma \vdash A$ is a *finite* tree built from the above inference rules whose root is $\Gamma \vdash A$. A judgment $\Gamma \vdash A$ is *provable* if there exists a proof of it. Observe that every proof has at its leaves either the rule for $\top$ or a conclusion from a hypothesis.

You may wonder what happened to negation. In intuitionistic propositional calculus, negation is defined in terms of implication and falsehood as

$$\neg A \;\equiv\; A \Rightarrow \bot \,.$$

Properties of negation are then derived from the rules for implication and falsehood, see Exercise 3.4.7

Let P be the set of all formulas of IPC, preordered by the relation

$$A \vdash B \,, \qquad (A, B \in P)$$

where we did not bother to label the hypothesis A. Clearly, it is the case that $A \vdash A$. To see that $\vdash$ is transitive, suppose Π_1 is a proof of $A \vdash B$ and Π_2 is a proof of $B \vdash C$. Then we can obtain a proof of $A \vdash C$ from a proof Π_2 of $B \vdash C$ by replacing in it each use of the hypothesis B by the proof Π_1 of $A \vdash B$. This is worked out in detail in the next two exercises.

Exercise 3.4.4. Prove the following statement by induction on the structure of the proof Π: if Π is a proof of $\Gamma, u : A, v : A \vdash B$ then there is a proof of $\Gamma, u : A \vdash B$.

Exercise 3.4.5. Prove the following statement by induction on the structure of the proof Π_2: if Π_1 is a proof of $\Gamma \vdash A$ and Π_2 is a proof of $\Gamma, u : A \vdash B$, then there is a proof of $\Gamma \vdash B$.

Let IPC be the poset reflection of the preorder $(P, \vdash)$. The elements of IPC are equivalence classes $[A]$ of formulas, where two formulas A and B are equivalent if both $A \vdash B$ and $B \vdash A$ are provable. The poset IPC is just the free Heyting algebra on countably many generators p_0, p_1, ...

Classical propositional calculus

Another look:

An element $x \in L$ of a lattice L is said to be *complemented* when there exists $y \in L$ such that

$$x \vee y = 1 , \qquad\qquad x \wedge y = 0 .$$

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* is a distributive lattice in which every element is complemented. In other words, a Boolean algebra B has the *complementation* operation $\neg$ which satisfies, for all $x \in B$,

$$x \wedge \neg x = 0 , \qquad\qquad x \vee \neg x = 1 . \tag{3.9}$$

The full subcategory of Lat consisting of Boolean algebras is denoted by Bool.

Exercise 3.4.6. 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 $\neg\neg x \leq x$ in general. An element x of a Heyting algebra for which $\neg\neg x = x$ is called a *regular* element.

Exercise 3.4.7. Derive the following properties of negation in a *Heyting* algebra:

$$\begin{aligned} x &\le \neg\neg x \ , \\ \neg x &= \neg\neg\neg x \ , \\ x \le y &\Rightarrow \neg y \le \neg x \ , \\ \neg\neg(x \wedge y) &= \neg\neg x \wedge \neg\neg y \ . \end{aligned}$$

Exercise 3.4.8. 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 3.4.9. 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 $x, y \in H_{\neg\neg}$:

$$\neg' x = \neg x \ , \qquad x \wedge' y = \neg\neg(x \wedge y) \ , \qquad x \vee' y = \neg\neg(x \vee y) \ .$$

The *classical propositional calculus (CPC)* is obtained from the intuitionistic propositional calculus by the addition of the logical rule known as *tertium non datur*, or the *law of excluded middle*:

$$\frac{}{\Gamma \vdash A \vee \neg A}$$

Alternatively, we could add the law known as *reductio ad absurdum*, or *proof by contradiction*:

$$\frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A} \ .$$

Identifying logically equivalent formulas of CPC, we obtain a poset CPC ordered by logical entailment. This poset is the *free Boolean algebra* on countably many generators. The construction of a free Boolean algebra can be performed just like described for the free Heyting algebra above. The equational axioms for a Boolean algebra are the axioms for a lattice (3.6), the distributive laws (3.7), and the complement laws (3.9).

Exercise* 3.4.10. Is CPC isomorphic to the Boolean algebra $\mathsf{IPC}_{\neg\neg}$ of the regular elements of IPC?

Exercise 3.4.11. 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:

$$\neg(x \wedge y) = \neg x \vee \neg y \ , \qquad \neg(x \vee y) = \neg x \wedge \neg y \ .$$

Together with $y \wedge \neg y = 0$, the De Morgan laws easily imply $y \vee \neg y = 1$. See [**?**, I.1.11].

Kripke semantics for IPC

We now prove the Kripke completeness of IPC, extending Theorem 3.3.4, namely:

Theorem 3.4.12 (Kripke completeness for IPC)**.** *Let K be a poset equipped with a* forcing relation *$k \Vdash p$ between elements $k \in K$ and propositional variables p, satisfying*

$$j \leq k,\ k \Vdash p \quad \textit{implies} \quad j \Vdash p. \tag{3.10}$$

Extend $\Vdash$ to all formulas ϕ in IPC by defining

$$\begin{array}{lll} k \Vdash \top & \textit{always}, & \\ k \Vdash \bot & \textit{never}, & \\ k \Vdash \phi \wedge \psi & \textit{iff} & k \Vdash \phi \textit{ and } k \Vdash \psi\,, \qquad (3.11) \\ k \Vdash \phi \vee \psi & \textit{iff} & k \Vdash \phi \textit{ or } k \Vdash \psi\,, \qquad (3.12) \\ k \Vdash \phi \Rightarrow \psi & \textit{iff} & \textit{for all } j \leq k, \textit{ if } j \Vdash \phi, \textit{ then } j \Vdash \psi\,. \end{array}$$

Finally, write $K \Vdash \phi$ if $k \Vdash \phi$ for all $k \in K$ (for all such relations $\Vdash$).

Then a propositional formulas ϕ is provable from the rules of deduction for IPC *(Definition 3.4.3) if, and only if, $K \Vdash \phi$ for all posets K. Briefly:*

$$\mathsf{PPC} \vdash \phi \quad \textit{iff} \quad K \Vdash \phi \textit{ for all } K.$$

Let us first see that we cannot simply reuse the proof from that theorem, because the downset (Yoneda) embedding that we used there

$$\downarrow\ : \mathsf{IPC} \hookrightarrow\ \downarrow\!(\mathsf{IPC}) \tag{3.13}$$

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 Joyal (cf. [**?**, **?**]). 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 $\mathsf{Bool}(\mathcal{B}, \mathbb{2})$, and the map $h : \mathcal{B} \rightarrowtail \mathcal{P}\mathsf{Bool}(\mathcal{B}, \mathbb{2})$ was given by $h(b) = \{F \,|\, b \in F\}$. Transposing $\mathcal{P}\mathsf{Bool}(\mathcal{B}, \mathbb{2}) \cong \mathbb{2}^{\mathsf{Bool}(\mathcal{B},\mathbb{2})}$ in the cartesian closed category Pos, we arrive at the (monotone) evaluation map

$$\mathsf{eval} : \mathsf{Bool}(\mathcal{B}, \mathbb{2}) \times \mathcal{B} \to \mathbb{2}. \tag{3.14}$$

Now recall that the category of Boolean algebras is full in the category DLat of distributive lattices,

$$\mathsf{Bool}(\mathcal{B}, \mathbb{2}) = \mathsf{DLat}(\mathcal{B}, \mathbb{2})\,.$$

For any Heyting algebra $\mathcal{H}$ (or indeed any distributive lattice), the Homset $\mathsf{DLat}(\mathcal{H}, 2)$, ordered pointwise, is isomorphic to the *poset* of all prime filters in $\mathcal{H}$ ordered by inclusion, by taking $f : \mathcal{H} \to 2$ to its (filter) kernel $f^{-1}\{1\} \subseteq \mathcal{H}$. In particular, the poset $\mathsf{DLat}(\mathcal{H}, 2)$ is no longer discrete when $\mathcal{H}$ is not Boolean, since a prime ideal in a Heyting algebra need not be maximal.

The transpose of the (monotone) evaluation map,

$$\mathsf{eval} : \mathsf{DLat}(\mathcal{H}, 2) \times \mathcal{H} \to 2. \tag{3.15}$$

will then be the (monotone) map

$$\epsilon\, : \mathcal{H} \longrightarrow 2^{\mathsf{DLat}(\mathcal{H},2)}, \tag{3.16}$$

which takes $p \in \mathcal{H}$ to the "evaluation at p" map $f \mapsto f(p) \in 2$, i.e.,

$$\epsilon_p(f) = f(p) \qquad \text{for } p \in \mathcal{H} \text{ and } f : \mathcal{H} \to 2\,.$$

As before, the poset $2^{\mathsf{DLat}(\mathcal{H},2)}$ (ordered pointwise) may be identified with the upsets in the poset $\mathsf{DLat}(\mathcal{H}, 2)$, ordered by inclusion, which recall from Example 3.4.2 is always a Heyting algebra. Thus, in sum, we have a monotone map,

$$\mathcal{H} \longrightarrow {\uparrow}\mathsf{DLat}(\mathcal{H}, 2)\,, \tag{3.17}$$

which generalizes the Stone representation from Boolean to Heyting algebras.

Theorem 3.4.13 (Joyal)**.** *Let $\mathcal{H}$ be a Heyting algebra. There is an injective Heyting homomorphism*

$$\mathcal{H} \rightarrowtail {\uparrow}J$$

into a Heyting algebra of upsets in a poset J.

Note that in this form, the theorem literally generalizes the Stone representation theorem, because when $\mathcal{H}$ is Boolean we can take J to be discrete, and then ${\uparrow}J \cong \mathsf{Pos}(J, 2) \cong \mathcal{P}J$ is Boolean, whence the Heyting embedding is also Boolean. The proof will again use the transposed evaluation map,

$$\epsilon : \mathcal{H} \longrightarrow {\uparrow}\mathsf{DLat}(\mathcal{H}, 2) \cong 2^{\mathsf{DLat}(\mathcal{H},2)}$$

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 3.4.14 (Birkhoff's 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 suffice 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 $I_0 \subseteq I_1 \subseteq \ldots$ 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 3.4.13. As in (3.17), let $J = \mathsf{DLat}(\mathcal{H}, \mathbb{2})$ be the poset of prime filters in $\mathcal{H}$, and consider the "evaluation" map (3.17),

$$\epsilon : \mathcal{H} \longrightarrow \mathbb{2}^{\mathsf{DLat}(\mathcal{H},\mathbb{2})} \cong \uparrow\mathsf{DLat}(\mathcal{H}, \mathbb{2})$$

given by $\epsilon(p) = \{F \mid p \in F \text{ prime}\}$.

Clearly $\epsilon(0) = \emptyset$ and $\epsilon(1) = \mathsf{DLat}(\mathcal{H}, \mathbb{2})$, and similarly for the other meets and joins, so ϵ is a lattice homomorphism. Moreover, if $p \neq q \in \mathcal{H}$ then, as in the proof of 2.6.8, we have that $\epsilon(p) \neq \epsilon(q)$, by the Prime Ideal Theorem (Lemma 3.4.14). Thus it just remains to show that

$$\epsilon(p \Rightarrow q) \;=\; \epsilon(p) \Rightarrow \epsilon(q)\,.$$

Unwinding the definitions, it suffices to show that, for all $f \in \mathsf{DLat}(\mathcal{H}, \mathbb{2})$,

$$f(p \Rightarrow q) = 1 \quad \text{iff} \quad \text{for all } g \geq f,\ g(p) = 1 \text{ implies } g(q) = 1. \tag{3.18}$$

Equivalently, for all prime filters $F \subseteq \mathcal{H}$,

$$p \Rightarrow q \in F \quad \text{iff} \quad \text{for all prime } G \supseteq F,\ p \in G \text{ implies } q \in G. \tag{3.19}$$

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 3.4.15. Give a Kripke countermodel to show that the Law of Excluded Middle $\phi \vee \neg\phi$ is not provable in IPC.

3.5 Frames and spaces

A poset $(P, \leq)$, viewed as a category, is *cocomplete* when it has suprema (least upper bounds) of arbitrary subsets. This is so because coequalizers in a poset always exist, and coproducts are precisely least upper bounds. Recall that the supremum of $S \subseteq P$ 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 the least element of P and $\bigvee P$ is the greatest element of P. Similarly, a poset is *complete* when it has infima (greatest lower bounds) of arbitrary subsets; the 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 3.5.1. *A poset is complete if, and only if, it is cocomplete.*

Proof. Infima and suprema are expressed in terms of each other as follows:

$$\begin{aligned} \bigwedge S &= \bigvee \left\{ y \in P \,\middle|\, \forall\, x : S \,.\, y \leq x \right\} , \\ \bigvee S &= \bigwedge \left\{ y \in P \,\middle|\, \forall\, x : S \,.\, x \leq y \right\} . \end{aligned}$$

□

Thus, we usually speak of *complete* posets only, even when we work with arbitrary suprema.

Suppose P is a complete poset. When is it cartesian closed? Being a complete poset, it has the terminal object, namely the greatest element $1 \in P$, 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 $z \in P$,

$$\frac{z \wedge x \leq y}{z \leq x \Rightarrow y} \,.$$

With the help of this adjunction we derive the *infinite distributive law*, for an arbitrary family $\left\{ y_i \in P \,\middle|\, i \in I \right\}$,

$$x \wedge \textstyle\bigvee_{i \in I} y_i = \bigvee_{i \in I} (x \wedge y_i) \tag{3.20}$$

as follows:

$$\begin{array}{c} x \wedge \bigvee_{i \in I} y_i \leq z \\ \hline\hline \bigvee_{i \in I} y_i \leq (x \Rightarrow z) \\ \hline\hline \forall\, i : I \,.\, (y_i \leq (x \Rightarrow z)) \\ \hline\hline \forall\, i : I \,.\, (x \wedge y_i \leq z) \\ \hline\hline \bigvee_{i \in I} (x \wedge y_i) \leq z \end{array}$$

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.

Conversely, suppose the distributive law (3.20) holds. Then we can *define* $x \Rightarrow y$ to be

$$(x \Rightarrow y) = \bigvee \left\{ z \in P \;\middle|\; x \wedge z \leq y \right\} \ . \tag{3.21}$$

The best way to show that $x \Rightarrow y$ is the exponential of x and y is to use the characterization of adjoints by counit, as in Proposition 1.5.5. In the case of $\wedge$ and $\Rightarrow$ this amounts to showing that, for all $x, y \in P$,

$$x \wedge (x \Rightarrow y) \leq y \ , \tag{3.22}$$

and that, for $z \in P$,

$$(x \wedge z \leq y) \Rightarrow (z \leq x \Rightarrow y) \ .$$

This implication follows directly from (3.5.7), and (3.22) follows from the distributive law:

$$x \wedge (x \Rightarrow y) = x \wedge \bigvee \left\{ z \in P \;\middle|\; x \wedge z \leq y \right\} = \bigvee \left\{ x \wedge z \;\middle|\; x \wedge z \leq y \right\} \leq y \ .$$

Complete cartesian closed posets are called *frames*.

Definition 3.5.2. A *frame* is a poset that is complete and cartesian closed, thus a frame is a complete Heyting algebra. Equivalently, a frame is a complete poset satisfying the (infinite) distributive law

$$x \wedge \bigvee_{i \in I} y_i = \bigvee_{i \in I}(x \wedge y_i) \ .$$

A *frame morphism* is a function $f : L \to M$ between frames that preserves finite infima and arbitrary suprema. The category of frames and frame morphisms is denoted by Frame.

Warning: a frame morphism need not preserve exponentials!

Example 3.5.3. 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 $\downarrow P$ is already known to be a Heyting algebra (Example 3.4.2), it is therefore also a frame. (Alternately, we can show that it is a frame by noting that the operations $\bigcup$ and $\bigcap$ satisfy the infinite distributive law, and then infer that it is a Heyting algebra.)

A monotone map $f : P \to Q$ between posets gives rise to a frame map

$$\downarrow f : \downarrow Q \longrightarrow \downarrow P \,,$$

as can be seen by recalling that $\downarrow P \cong 2^P$ as posets. Note that as a (co)limit preserving functor on complete posets, $2^f : 2^Q \longrightarrow 2^P$ has both left and right adjoints. These functors are usually written $f_! \dashv f^* \dashv f_*$. Although it does not in general preserve Heyting implications $S \Rightarrow T$, the monotone map $\downarrow f : \downarrow Q \longrightarrow \downarrow P$ is indeed a morphism of frames. We therefore have a contravariant functor

$$\downarrow(-) : \ \mathsf{Pos} \to \mathsf{Frame}^{\mathsf{op}}. \tag{3.23}$$

Example 3.5.4. 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 $f^* : \mathcal{O}Y \to \mathcal{O}X$ 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 $f^* : \mathcal{O}Y \to \mathcal{O}X$.

The category $\mathsf{Frame}^{\mathsf{op}}$ is called the category of *locales* and is denoted by Loc. When we think of a frame as an object of Loc we call it a locale.

Example 3.5.5. Let P be a poset and define a topology on the elements of P by defining the opens to be the upsets,

$$\mathcal{O}P = \uparrow P \cong \mathsf{Pos}(P, \mathbb{2}).$$

These open sets are not only closed under arbitrary unions and finite intersections, but also under *arbitrary* intersections. Such a topological space is said to be an *Alexandrov space.*

Exercise* 3.5.6. This exercise is meant for students with some background in 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 3.5.7. A *topological model* of IPC is a space X and an interpretation of formulas,

$$[\![-]\!] : \mathsf{IPC} \to \mathcal{O}X ,$$

satisfying the conditions:

$$
\begin{aligned}
[\![\top]\!] &= X \\
[\![\bot]\!] &= \emptyset \\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cap [\![\psi]\!] \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!] \\
[\![\phi \Rightarrow \psi]\!] &= [\![\phi]\!] \Rightarrow [\![\psi]\!]
\end{aligned}
$$

The Heyting implication $[\![\phi]\!] \Rightarrow [\![\psi]\!]$ in $\mathcal{O}X$, is defined in (3.5.7) as

$$[\![\phi]\!] \Rightarrow [\![\psi]\!] \;=\; \bigcup \big\{ U \in \mathcal{O}X \;\big|\; U \wedge [\![\phi]\!] \leq [\![\psi]\!] \big\} \;.$$

Joyal's representation theorem 3.4.13 easily implies that IPC is sound and complete with respect to topological semantics.

Corollary 3.5.8. *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 Alexandrov topology on the upsets of prime ideals in the Heyting algebra IPC □

Exercise 3.5.9. Give a topological countermodel to show that the Law of Double Negation $\neg\neg\phi \Rightarrow \phi$ is not provable in IPC.

3.6 Proper CCCs

We begin by reviewing some important examples of cartesian closed categories that are *not posets*, most of which have already been discussed.

Example 3.6.1. 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 3.6.2. 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 1.6). 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, too, and is therefore a full inclusion of CCCs.

Example 3.6.3. 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. It is worth noting that, unlike the previous cases, the (limit preserving) forgetful functor $U : \mathsf{Poset} \to \mathsf{Set}$ does *not* preserve exponentials; in general $U(Q^P) \subseteq (UQ)^{UP}$ is a *proper* subset.

Exercise 3.6.4. 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 3.6.5. *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 3.6.6. *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 3.6.7. Which of the examples follows from which proposition?

Example 3.6.8. 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 3.6.9. 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 3.6.10. 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 also 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 [**?**].

3.7 Simply typed λ-calculus

The λ-calculus is the abstract theory of functions, just like group theory is the 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 abstractions are nested:

$$\lambda x.\, t\ .$$

This operation is called λ-*abstraction*. For example, $\lambda x.\,\lambda y.\,(x+y)$ is the function which maps an argument a to the function $\lambda y.\,(a+y)$, which maps an argument b ... .

In an expression $\lambda x.\, t$ the variable x is said to be *bound* in t.

Remark 3.7.1. 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 since its beginnings, 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 will occupy the remainder of this chapter.

Remark 3.7.2. There are two kinds of λ-calculus, the *typed* and the *untyped* one. 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. 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 f, a 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 $t \in A$. We will concentrate on the typed λ-calculus.

We now give a precise 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:

$$\begin{aligned} \text{Basic type } \mathtt{B} &::= \mathtt{B}_0 \mid \mathtt{B}_1 \mid \mathtt{B}_2 \cdots \\ \text{Simple type } A &::= \mathtt{B} \mid A_1 \times A_2 \mid A_1 \to A_2. \end{aligned}$$

Function types associate to the right:

$$A \to B \to C \equiv A \to (B \to C)\,.$$

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:

$$\begin{aligned} \text{Variable } v &::= x \mid y \mid z \mid \cdots \\ \text{Constant } c &::= \mathtt{c}_1 \mid \mathtt{c}_2 \mid \cdots \\ \text{Term } t &::= v \mid c \mid * \mid \langle t_1, t_2\rangle \mid \mathtt{fst}\,t \mid \mathtt{snd}\,t \mid t_1\,t_2 \mid \lambda x : A\,.\,t \end{aligned}$$

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 *free variables* $\mathsf{FV}(t)$ of a term t are computed as follows:

$$\begin{aligned}
\mathsf{FV}(x) &= \{x\} \qquad \text{if } x \text{ is a variable} \\
\mathsf{FV}(a) &= \emptyset \qquad \text{if } a \text{ is a basic constant} \\
\mathsf{FV}(\langle u, t\rangle) &= \mathsf{FV}(u) \cup \mathsf{FV}(t) \\
\mathsf{FV}(\mathtt{fst}\, t) &= \mathsf{FV}(t) \\
\mathsf{FV}(\mathtt{snd}\, t) &= \mathsf{FV}(t) \\
\mathsf{FV}(u\, t) &= \mathsf{FV}(u) \cup \mathsf{FV}(t) \\
\mathsf{FV}(\lambda x.\, t) &= \mathsf{FV}(t) \setminus \{x\} \ .
\end{aligned}$$

If $x_1, \ldots, x_n$ are *distinct* variables and $A_1, \ldots, A_n$ 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. Context 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 c_i has a uniquely determined type C_i,

$$\frac{}{\Gamma \mid \mathsf{c}_i : C_i}$$

The type of a variable is determined by the context:

$$\frac{}{x_1 : A_1, \ldots, x_i : A_i, \ldots, x_n : A_n \mid x_i : A_i}\ (1 \le i \le n)$$

The constant $*$ has type $\mathtt{1}$:

$$\frac{}{\Gamma \mid * : \mathtt{1}}$$

The typing rules for pairs and projections are:

$$\frac{\Gamma \mid u : A \qquad \Gamma \mid t : B}{\Gamma \mid \langle u, t\rangle : A \times B} \qquad \frac{\Gamma \mid t : A \times B}{\Gamma \mid \mathtt{fst}\, t : A} \qquad \frac{\Gamma \mid t : A \times B}{\Gamma \mid \mathtt{snd}\, t : B}$$

The typing rules for application and λ-abstraction are:

$$\frac{\Gamma \mid t : A \to B \qquad \Gamma \mid u : A}{\Gamma \mid t\, u : B} \qquad \frac{\Gamma, x : A \mid t : B}{\Gamma \mid (\lambda x : A \,.\, t) : A \to B}$$

Lastly, we have *equations* between terms; for terms of type A in context Γ,

$$\Gamma \mid u : A \,, \qquad \Gamma \mid t : B \,,$$

the judgment that they are equal is written as

$$\Gamma \mid u = t : A \,.$$

Note that u 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:

1. Equality is an equivalence relation:

$$\frac{}{\Gamma \mid t = t : A} \qquad \frac{\Gamma \mid t = u : A}{\Gamma \mid u = t : A} \qquad \frac{\Gamma \mid t = u : A \qquad \Gamma \mid u = v : A}{\Gamma \mid t = v : A}$$

2. The weakening rule:

$$\frac{\Gamma \mid u = t : A}{\Gamma, x : B \mid u = t : A}$$

3. Unit type:

$$\frac{}{\Gamma \mid t = * : \mathtt{1}}$$

4. Equations for product types:

$$\frac{\Gamma \mid u = v : A \qquad \Gamma \mid s = t : B}{\Gamma \mid \langle u, s\rangle = \langle v, t\rangle : A \times B}$$

$$\frac{\Gamma \mid s = t : A \times B}{\Gamma \mid \mathtt{fst}\, s = \mathtt{fst}\, t : A} \qquad \frac{\Gamma \mid s = t : A \times B}{\Gamma \mid \mathtt{snd}\, s = \mathtt{snd}\, t : A}$$

$$\frac{}{\Gamma \mid t = \langle \mathtt{fst}\, t, \mathtt{snd}\, t\rangle : A \times B}$$

$$\frac{}{\Gamma \mid \mathtt{fst}\, \langle u, t\rangle = u : A} \qquad \frac{}{\Gamma \mid \mathtt{snd}\, \langle u, t\rangle = t : A}$$

5. 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.

Apart from the above rules for equality we might want to impose additional equations. In this case we do not speak of a λ-calculus but rather of a *λ-theory*. Thus, a λ-theory $\mathbb{T}$ is given by a set of basic types, a set of basic constants, and a set of *equations* of the form

$$\Gamma \mid u = t : A \,.$$

We summarize the preceding definitions.

Definition 3.7.3. A *simply-typed λ-calculus* is given by a set of *basic types* and a set of *basic constants* together with their types. *A simply-typed λ-theory* is a simply-typed λ-calculus together with a set of equations.

We use letters $\mathbb{S}$, $\mathbb{T}$, $\mathbb{U}$, ... to denote theories.

Example 3.7.4. The theory of a group is a simply-typed λ-theory. It has one basic type $\mathtt{G}$ and three basic constant, the unit $\mathtt{e}$, the inverse $\mathtt{i}$, and the group operation $\mathtt{m}$,

$$\mathtt{e} : \mathtt{G}\,, \qquad \mathtt{i} : \mathtt{G} \to \mathtt{G}\,, \qquad \mathtt{m} : \mathtt{G} \times \mathtt{G} \to \mathtt{G}\,,$$

with the following equations:

$$\begin{aligned}
x : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{e}\rangle = x : \mathtt{G}\\
x : \mathtt{G} &\mid \mathtt{m}\langle \mathtt{e}, x\rangle = x : \mathtt{G}\\
x : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{i}\,x\rangle = \mathtt{e} : \mathtt{G}\\
x : \mathtt{G} &\mid \mathtt{m}\langle \mathtt{i}\,x, x\rangle = \mathtt{e} : \mathtt{G}\\
x : \mathtt{G}, y : \mathtt{G}, z : \mathtt{G} &\mid \mathtt{m}\langle x, \mathtt{m}\langle y, z\rangle\rangle = \mathtt{m}\langle \mathtt{m}\langle x, y\rangle, z\rangle : \mathtt{G}
\end{aligned}$$

These are just the familiar axioms for a group.

Example 3.7.5. In general, any algebraic theory $\mathbb{A}$ 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 the terms of the corresponding λ-theory in a straightforward manner. For every axiom $t = u$ of $\mathbb{A}$ the corresponding axiom in the λ-theory is

$$x_1 : \mathtt{A}, \ldots, x_n : \mathtt{A} \mid t = u : \mathtt{A}$$

where $x_1, \ldots, x_n$ are the variables occurring in t and u.

Example 3.7.6. 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 constant, 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 3.7.7. An example of a λ-theory is readily found in the theory of programming languages. The mini-programming language PCF is a simply-typed λ-calculus with a basic type $\mathtt{nat}$ for natural numbers, and a basic type $\mathtt{bool}$ of Boolean values,

$$\text{Basic type } \mathtt{B} ::= \mathtt{nat\ type} \mid \mathtt{bool\ type}.$$

There are basic constants zero $\mathtt{0}$, successor $\mathtt{succ}$, the Boolean constants $\mathtt{true}$ and $\mathtt{false}$, comparison with zero $\mathtt{iszero}$, 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 3.7.8. Another example of a λ-theory is the *theory of a reflexive type*. This theory has one basic type $\mathtt{D}$ 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

$$f : \mathtt{D} \to \mathtt{D} \mid \mathtt{r}\,(\mathtt{s}\,f) = f : \mathtt{D} \to \mathtt{D} \tag{3.24}$$

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

$$x : \mathtt{D} \mid \mathtt{s}\,(\mathtt{r}\,x) = x : \mathtt{D} \tag{3.25}$$

which implies that $\mathtt{D}$ is isomorphic to $\mathtt{D} \to \mathtt{D}$.

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:

$$\frac{\Gamma \mid s = t \qquad \Gamma \mid u = v}{\Gamma \mid s\, u = t\, v} \qquad \frac{\Gamma, x \mid t = u}{\Gamma \mid \lambda x.\, t = \lambda x.\, u}$$

$$\frac{}{\Gamma \mid (\lambda x.\, t)u = t[u/x]} \tag{β-rule}$$

$$\frac{}{\Gamma \mid \lambda x.\, (t\, x) = t} \quad \text{if } x \notin \mathsf{FV}(t) \tag{η-rule}$$

The untyped λ-calculus can be translated into the theory of a reflexive type from Example 3.7.8. 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

$$\Gamma \mid u = t \tag{3.26}$$

is translated to the judgment

$$\Gamma^* \mid u^* = t^* : \mathtt{D}\;. \tag{3.27}$$

Exercise* 3.7.9. Prove that if equation (3.26) is provable then equation (3.27) is provable as well. Identify precisely at which point in your proof you need to use equations (3.24) and (3.25). Does provability of (3.27) imply provability of (3.26)?

3.8 Interpretation of λ-calculus in CCCs

We now consider semantic aspects of λ-calculus and λ-theories. Suppose $\mathbb{T}$ is a λ-calculus and $\mathcal{C}$ is a cartesian closed category. An *interpretation* $[\![-]\!]$ of $\mathbb{T}$ in $\mathcal{C}$ is given by the following data:

1. For every basic type A in $\mathbb{T}$ an object $[\![A]\!] \in \mathcal{C}$. The interpretation is extended to all types by

$$[\![\mathtt{1}]\!] = \mathsf{1}\;, \qquad [\![A \times B]\!] = [\![A]\!] \times [\![B]\!]\;, \qquad [\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}\;.$$

2. For every basic constant c of type A a morphism $[\![c]\!] : \mathsf{1} \to [\![A]\!]$.

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

$$[\![A_1]\!] \times \cdots \times [\![A_n]\!]\;,$$

and the empty context is interpreted as the terminal object $\mathsf{1}$. A typing judgment

$$\Gamma \mid t : A$$

is interpreted as a morphism

$$[\![\Gamma \mid t : A]\!] : [\![\Gamma]\!] \to [\![A]\!]\;.$$

The interpretation is defined inductively by the following rules:

1. The i-th variable is interpreted as the i-th projection,

$$[\![x_0 : A_0, \ldots, x_n : A_n \mid x_i : A_i]\!] = \pi_i : [\![\Gamma]\!] \to [\![A_i]\!]\;.$$

2. A basic constant $c : A$ in context Γ is interpreted as the composition

$$[\![\Gamma]\!] \xrightarrow{!_{[\![\Gamma]\!]}} \mathsf{1} \xrightarrow{[\![c]\!]} [\![A]\!]$$

3. The interpretation of projections and pairs is

$$\begin{aligned}
[\![\Gamma \mid \langle t, u\rangle : A \times B]\!] &= \langle [\![\Gamma \mid t : A]\!], [\![\Gamma \mid u : B]\!]\rangle : [\![\Gamma]\!] \to [\![A]\!] \times [\![B]\!] \\
[\![\Gamma \mid \mathtt{fst}\, t : A]\!] &= \pi_0 \circ [\![\Gamma \mid t : A \times B]\!] : [\![\Gamma]\!] \to [\![A]\!] \\
[\![\Gamma \mid \mathtt{snd}\, t : A]\!] &= \pi_1 \circ [\![\Gamma \mid t : A \times B]\!] : [\![\Gamma]\!] \to [\![B]\!]\,.
\end{aligned}$$

4. The interpretation of application and λ-abstraction is

$$\begin{aligned}
[\![\Gamma \mid t\, u : B]\!] &= e \circ \langle [\![\Gamma \mid t : A \to B]\!], [\![\Gamma \mid u : A]\!]\rangle : [\![\Gamma]\!] \to [\![B]\!] \\
[\![\Gamma \mid \lambda x : A \,.\, t : A \to B]\!] &= ([\![\Gamma, x : A \mid t : B]\!])^{\sim} : [\![\Gamma]\!] \to [\![B]\!]^{[\![A]\!]}
\end{aligned}$$

   where $e : [\![A \to B]\!] \times [\![A]\!] \to [\![B]\!]$ is the evaluation morphism for $[\![B]\!]^{[\![A]\!]}$ and $([\![\Gamma, x : A \mid t : B]\!])^{\sim}$ is the transpose of the morphism

$$[\![\Gamma, x : A \mid t : B]\!] : [\![\Gamma]\!] \times [\![A]\!] \to [\![B]\!]\,.$$

An interpretation of the λ-calculus of a theory $\mathbb{T}$ is a *model* of the theory if it satisfies all axioms of $\mathbb{T}$. This means that, for every axiom $\Gamma \mid t = u : A$, the interpretations of u and t coincide as arrows in $\mathcal{C}$,

$$[\![\Gamma \mid u : A]\!] = [\![\Gamma \mid t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!].$$

It follows that all equations provable in $\mathbb{T}$ are satisfied in the model, by the following fact.

Proposition 3.8.1 (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 $\Gamma \mid t = u : A$ that is provable from the axioms of $\mathbb{T}$, we have*

$$[\![\Gamma \mid u : A]\!] = [\![\Gamma \mid t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]\,.$$

Briefly, for all $\mathbb{T}$-models $[\![-]\!]$,

$$\mathbb{T} \vdash (\Gamma \mid t = u : A) \quad \textit{implies} \quad [\![-]\!] \models (\Gamma \mid t = u : A)\,.$$

Remark 3.8.2 (Inhabitation)**.** There is another notion of provability for the λ-calculus, related to the Curry-Howard correspondence of section 3.1, relating it to propositional logic. If we regard types as "propositions" rather than structures, and terms as "proofs" rather than operations, 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 $s = t : A$. 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 a *provable.* In this sense, there is another notion of soundness as follows.

Proposition 3.8.3 (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 closed type A that is inhabited in $\mathbb{T}$ by a closed term, $\vdash a : A$, there is a corresponding point*

$$[\![a]\!] : 1 \to [\![A]\!]\,,$$

in $\mathcal{C}$. Briefly, for all $\mathbb{T}$-models $[\![-]\!]$,

$$\vdash a : A \quad \textit{implies} \quad [\![a]\!] : 1 \to [\![A]\!]\,.$$

This follows immediately from the fact that $[\![\cdot]\!] = 1$ for $\Gamma = \cdot$ the empty context.

Example 3.8.4.

1. A model of an algebraic theory $\mathbb{A}$, extended to a λ-theory $\mathbb{A}_\lambda$ as in Example 3.7.5, taken in a CCC $\mathcal{C}$, is just a model of the algebraic theory $\mathbb{A}$ in the underlying finite product category $|\mathcal{C}|$ of $\mathcal{C}$. A difference, however, is that in defining the *category of models*
$$\mathsf{Mod}_\times(\mathbb{A}, |\mathcal{C}|)$$
we can take all homomorphism of models the algebraic theory 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 $[\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}$.

2. A model of the theory of a reflexive type, Example 3.7.8, in Set must be the one-element 1 (prove this!). Fortunately, the exponentials in categories of presheaves are *not* computed pointwise - otherwise it would follow that there would be no non-trivial models at all in small categories! That there are such non-trivial models is an important fact in the semantics of programming languages and the subject called *domain theory*. A basic paper in which this is shown is [**?**].

3. A model of a propositional theory $\mathbb{T}$, regarded as a λ-theory, in a CCC poset P is 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, i.e. $[\![\phi_1]\!] = [\![\phi_2]\!] = ... = 1 \in P$.

3.9 Functorial semantics

In Section **??** we saw that algebraic theories can be viewed as categories, cf. Definition **??**, and models as functors, cf. Definition **??**, and we arranged this categorical analysis of the traditional relationship between syntax and sematics into the framework that we called *functorial semantics*. The same can be done with λ-theories and their models in CCCs. The first step is to build a *syntactic category* $\mathcal{C}_\mathbb{T}$ from a λ-theory $\mathbb{T}$. This is done as follows:

- The objects of $\mathcal{C}_\mathbb{T}$ are the types of $\mathbb{T}$.

- Morphisms $A \to B$ are terms in context

$$[x : A \mid t : B] ,$$

where two such terms $x : A \mid t : B$ and $x : A \mid u : B$ represent the same morphism when $\mathbb{T}$ proves $x : A \mid t = u : B$.

- 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 3.9.1. *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 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

$$x : A \mid * : \mathtt{1} .$$

This morphism is unique because

$$\Gamma \mid t = \star : \mathtt{1}$$

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

$$p : A \times B \mid \mathtt{fst}\, p : A , \qquad\qquad p : A \times B \mid \mathtt{snd}\, p : B .$$

Given morphisms

$$z : C \mid t : A , \qquad\qquad z : C \mid u : B ,$$

the term

$$z : C \mid \langle t, u \rangle : A \times B$$

represents the unique morphism satisfying

$$z : C \mid \mathtt{fst}\, \langle t, u \rangle = t : A , \qquad\qquad z : C \mid \mathtt{snd}\, \langle t, u \rangle = u : B .$$

Indeed, if $\mathtt{fst}\, s = t$ and $\mathtt{snd}\, s = u$ then

$$s = \langle \mathtt{fst}\, s, \mathtt{snd}\, s \rangle = \langle t, u \rangle .$$

- The exponential of objects A and B is the type $A \to B$ with the evaluation morphism

$$p : (A \to B) \times A \mid (\mathtt{fst}\, p)(\mathtt{snd}\, p) : B \ .$$

The transpose of the morphism $p : C \times A \mid t : B$ is

$$z : C \mid \lambda x : A \,.\, (t[\langle z, x\rangle/p]) : A \to B \ .$$

Showing that this is the transpose of t amounts to

$$(\lambda x : A \,.\, (t[\langle \mathtt{fst}\, p, x\rangle/p]))(\mathtt{snd}\, p) = t[\langle \mathtt{fst}\, p, \mathtt{snd}\, p\rangle/p] = t[p/p] = t \ ,$$

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

$$(s[\mathtt{fst}\, p/z])(\mathtt{snd}\, p) = t$$

is equal to $\lambda x : A \,.\, (t[\langle z, x\rangle/p])$. First observe that

$$t[\langle z, x\rangle/p] = (s[\mathtt{fst}\, p/z])(\mathtt{snd}\, p)[\langle z, x\rangle/p] = \\ (s[\mathtt{fst}\, \langle z, x\rangle/z])(\mathtt{fst}\, \langle z, x\rangle) = (s[z/z])\, x = s\, x \ .$$

Therefore,

$$\lambda x : A \,.\, (t[\langle z, x\rangle/p]) = \lambda x : A \,.\, (s\, x) = s \ ,$$

as required.

□

The syntactic category allows us to "redefine" models as functors. More precisely, we have the following.

Lemma 3.9.2. *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*

$$M(A) = [\![A]\!], \quad M(c) = [\![c]\!], \tag{3.28}$$

for all basic types A and basic constants c. Moreover, M is unique up to a unique isomorphism of CCC functors, in the sense that given another model N satisfying (3.28)*, 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 now have the usual functorial semantics theorem:

Theorem 3.9.3. *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*

$$\mathsf{Mod}_{\lambda}(\mathbb{T}, \mathcal{C}) \simeq \mathsf{CCC}(\mathcal{C}_{\mathbb{T}}, \mathcal{C}), \tag{3.29}$$

naturally in $\mathcal{C}$.

Proof. Note that the categories involved in (3.29) are actually groupoids, as discussed in example 3.8.4(1). The only thing remaining to show is that given a model $[\![-]\!]^M$ in a CCC $\mathcal{C}$ and a CCC functor $f : \mathcal{C} \to \mathcal{D}$, there is an induced model $[\![-]\!]^{fM}$ in $\mathcal{D}$, given by the interpretation $[\![A]\!]^{fM} = f[\![A]\!]^M$. This is straigtforward, just as for algebraic theories. □

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

$$\begin{aligned} \mathbb{T} \vdash (\Gamma \mid u = t : A) &\iff [\Gamma \mid u : A] = [\Gamma \mid t : A] \\ &\iff [-] \models \Gamma \mid u = t : A\,. \end{aligned}$$

For the record, we therefore have shown:

Proposition 3.9.4. *For any λ-theory $\mathbb{T}$,*

$\mathbb{T} \vdash (\Gamma \mid t = u : A)$ *if, and only if,* $[-] \models (\Gamma \mid t = u : A)$ *for the syntactic model* $[-]$.

Of course, the syntactic model $[-]$ is the one associated under (3.29) 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 3.8.1).

Corollary 3.9.5. *For any λ-theory $\mathbb{T}$,*

$\mathbb{T} \vdash (\Gamma \mid t = u : A)$ *if, and only if,* $M \models (\Gamma \mid t = u : A)$ *for every CCC model M.*

Moreover, a closed type A is inhabited $\vdash a : A$ if, and only if, there is a point $1 \to [\![A]\!]$ in every model M.

3.10 The internal language of a CCC

We can take the correspondence between λ-theories and CCCs one step further and organize the former into a category, which is then equivalent to that of the latter. For this we first need to define a suitable notion of *morphism of theories*. A *translation* $\tau : \mathbb{T} \to \mathbb{U}$ of a λ-theory $\mathbb{T}$ into a λ-theory $\mathbb{U}$ is given by the following data:

1. For each basic type A in $\mathbb{T}$ a type τA in $\mathbb{U}$. The translation is then extended to all types by the rules

$$\tau \mathtt{1} = \mathtt{1}\ , \qquad \tau(A \times B) = \tau A \times \tau B\ , \qquad \tau(A \to B) = \tau A \to \tau B\ .$$

2. For each basic constant c of type A in $\mathbb{A}$ a term τc of type τA in $\mathbb{U}$. The translation of terms is then extended to all terms by the rules

$$\begin{aligned} \tau(\mathtt{fst}\, t) &= \mathtt{fst}\,(\tau t)\ , & \tau(\mathtt{snd}\, t) &= \mathtt{snd}\,(\tau t)\ , \\ \tau\langle t, u\rangle &= \langle \tau t, \tau u\rangle\ , & \tau(\lambda x : A \,.\, t) &= \lambda x : \tau A \,.\, \tau t\ , \\ \tau(t\, u) &= (\tau t)(\tau u)\ , & \tau x &= x \quad \text{(if } x \text{ is a variable)}\ . \end{aligned}$$

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 $\mathbb{T}$: if $\Gamma \mid t = u : A$ is an axiom of $\mathbb{T}$ then $\mathbb{U}$ proves $\tau\Gamma \mid \tau t = \tau u : \tau A$. It then follows that all equations proved by $\mathbb{T}$ are translated to valid equations in $\mathbb{U}$.

A moment's consideration shows that a translation $\tau : \mathbb{T} \to \mathbb{U}$ is the same thing as a model of $\mathbb{T}$ in $\mathcal{C}_{\mathbb{U}}$, despite being specified entirely syntactically. Clearly, λ-theories and translations between them 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. It corresponds to the canonical interpretation of $\mathbb{T}$ in $\mathcal{C}_{\mathbb{T}}$.

Definition 3.10.1. $\lambda\mathsf{Thr}$ is the category whose objects are λ-theories and morphisms are translations between them.

Let $\mathcal{C}$ be a small cartesian closed category. There is a λ-theory $\mathbb{L}(\mathcal{C})$ that corresponds to $\mathcal{C}$, called the *internal language of* $\mathcal{C}$, 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

$$x : \ulcorner A \urcorner \mid \ulcorner \mathtt{1}_A \urcorner\, x = x : \ulcorner A \urcorner\ .$$

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
$$\mathtt{T} : \mathtt{1} \to \ulcorner \mathtt{1} \urcorner\,,$$
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:
$$\begin{aligned} u : \ulcorner \mathtt{1} \urcorner &\mid \mathtt{T} * = u : \ulcorner \mathtt{1} \urcorner \\ z : \ulcorner A \times B \urcorner &\mid \mathtt{P}_{A,B} \langle \ulcorner \pi_0 \urcorner z, \ulcorner \pi_1 \urcorner z \rangle = z : \ulcorner A \times B \urcorner \\ w : \ulcorner A \urcorner \times \ulcorner B \urcorner &\mid \langle \ulcorner \pi_0 \urcorner (\mathtt{P}_{A,B} w), \ulcorner \pi_1 \urcorner (\mathtt{P}_{A,B} w) \rangle = w : \ulcorner A \urcorner \times \ulcorner B \urcorner \\ f : \ulcorner B^A \urcorner &\mid \mathtt{E}_{A,B}(\lambda x : \ulcorner A \urcorner .\, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle f, x \rangle))) = f : \ulcorner B^A \urcorner \\ f : \ulcorner A \urcorner \to \ulcorner B \urcorner &\mid \lambda x : \ulcorner A \urcorner .\, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle (\mathtt{E}_{A,B} f), x \rangle)) = f : \ulcorner A \urcorner \to \ulcorner B \urcorner \end{aligned}$$

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 $\ulcorner \mathtt{1} \urcorner \cong \mathtt{1}$, $\ulcorner A \times B \urcorner \cong \ulcorner A \urcorner \times \ulcorner B \urcorner$, 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
$$x : A \mid t : B\,, \qquad\qquad y : B \mid u : A\,,$$
such that $\mathbb{T}$ proves
$$x : A \mid u[t/y] = x : A\,, \qquad\qquad y : B \mid t[u/x] = y : B\,.$$
Furthermore, an *equivalence of theories* $\mathbb{T}$ and $\mathbb{U}$ is a pair of translations
$$\begin{tikzcd} \mathbb{T} \arrow[r, bend left, "\tau"] & \mathbb{U} \arrow[l, bend left, "\sigma"] \end{tikzcd}$$
such that, for any type A in $\mathbb{T}$ and any type B in $\mathbb{U}$,
$$\sigma(\tau A) \cong A\,, \qquad\qquad \tau(\sigma B) \cong B\,.$$

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 $\mathbb{T}$ to a small cartesian closed category $\mathcal{C}_{\mathbb{T}}$. This extends to a functor

$$\mathcal{C} : \lambda\mathsf{Thr} \to \mathsf{CCC} ,$$

because a translation $\tau : \mathbb{T} \to \mathbb{U}$ induces a functor $\mathcal{C}_\tau : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{U}}$ in an obvious way: a basic type $A \in \mathcal{C}_{\mathbb{T}}$ is mapped to the object $\tau A \in \mathcal{C}_{\mathbb{U}}$, 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 3.10.2. *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". This means that for any $\mathcal{C} \in \mathsf{CCC}$ there is an equivalence of catgories*

$$\mathcal{C} \simeq \mathcal{C}_{\mathbb{L}(\mathcal{C})} ,$$

and for any $\mathbb{T} \in \lambda\mathsf{Thr}$ there is an equivalence of theories

$$\mathbb{T} \simeq \mathbb{L}(\mathcal{C}_{\mathbb{T}}) .$$

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

$$\eta_{\mathcal{C}} A = \ulcorner A \urcorner , \qquad \eta_{\mathcal{C}} f = (x : \ulcorner A \urcorner \mid \ulcorner f \urcorner x : \ulcorner B \urcorner) .$$

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 $\mathbb{T}$, we define a translation $\tau_{\mathbb{T}} : \mathbb{T} \to \mathbb{L}(\mathcal{C}_{\mathbb{T}})$. For a basic type A let

$$\tau_{\mathbb{T}} A = \ulcorner A \urcorner .$$

The translation $\tau_{\mathbb{T}} c$ of a basic constant c of type A is

$$\tau_{\mathbb{T}} c = \ulcorner x : \mathtt{1} \mid c : \tau_{\mathbb{T}} A \urcorner .$$

In the other direction we define a translaton $\sigma_{\mathbb{T}} : \mathbb{L}(\mathcal{C}_{\mathbb{T}}) \to \mathbb{T}$ as follows. If $\ulcorner A \urcorner$ is a basic type in $\mathbb{L}(\mathcal{C}_{\mathbb{T}})$ then

$$\sigma_{\mathbb{T}} \ulcorner A \urcorner = A ,$$

and if $\ulcorner x : A \mid t : B \urcorner$ is a basic constant of type $\ulcorner A \urcorner \to \ulcorner B \urcorner$ then

$$\sigma_{\mathbb{T}} \ulcorner x : A \mid t : B \urcorner = \lambda x : A \,.\, t .$$

The basic constants $\mathtt{T}$, $\mathtt{P}_{A,B}$ and $\mathtt{E}_{A,B}$ are translated by $\sigma_{\mathbb{T}}$ into

$$\begin{aligned} \sigma_{\mathbb{T}} \, \mathtt{T} &= \lambda x : \mathtt{1} \,.\, x , \\ \sigma_{\mathbb{T}} \, \mathtt{P}_{A,B} &= \lambda p : A \times B \,.\, p , \\ \sigma_{\mathbb{T}} \, \mathtt{E}_{A,B} &= \lambda f : A \to B \,.\, f . \end{aligned}$$

If A is a type in $\mathbb{T}$ then $\sigma_{\mathbb{T}}(\tau_{\mathbb{T}} A) = A$. For the other direction, we would like to show, for any type X in $\mathbb{L}(\mathcal{C}_{\mathbb{T}})$, that $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} X) \cong X$. We prove this by induction on the structure of type X:

1. If $X = \mathtt{1}$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \mathtt{1}) = \mathtt{1}$.
2. If $X = \ulcorner A \urcorner$ is a basic type then A is a type in $\mathbb{T}$. We proceed by induction on the structure of A:
   (a) If $A = \mathtt{1}$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner \mathtt{1} \urcorner) = \mathtt{1}$. 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 $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner A \urcorner) = \ulcorner A \urcorner$.
   (c) If $A = B \times C$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner B \times C \urcorner) = \ulcorner B \urcorner \times \ulcorner C \urcorner$. But we know $\ulcorner B \urcorner \times \ulcorner C \urcorner \cong \ulcorner B \times C \urcorner$ via the constant $\mathtt{P}_{A,B}$.
   (d) The case $A = B \to C$ is similar.
3. If $X = Y \times Z$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}}(Y \times Z)) = \tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Y) \times \tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Z)$. By induction hypothesis, $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Y) \cong Y$ and $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Z) \cong Z$, from which we easily obtain

$$\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Y) \times \tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Z) \cong Y \times Z .$$

4. The case $X = Y \to Z$ is similar.

□

Exercise 3.10.3. In the previous proof 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 $\eta : 1_{\mathsf{CCC}} \Longrightarrow \mathcal{C} \circ \mathbb{L}$. Can you say anything about naturality of the translations $\tau_{\mathbb{T}}$ and $\sigma_{\mathbb{T}}$? What would it even mean for a translation to be natural?

Remark 3.10.4. Discussion of untyped λ-calculus: we do not know that the syntactic construction is non-trivial. But existence of non-trivial models tells us that it is not (which implies a suitable notion of consistency of untyped λ-calculus).

Give an untyped model satisfying β-reduction. Refer to literature for $\beta\eta$-models.

Remark 3.10.5. Adding coproducts $0, A + B$, also for presheaf models.

3.11 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 3.8.2; 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 3.1, 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 3.11.1. *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 3.11.2. Prove Lemma 3.11.1.

Corollary 3.11.3. *The syntactic category $\mathsf{PPC}(p_1, p_2, ..., p_n)$ of the positive propositional calculus on n propositional variables is the poset reflection 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 $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$ is the free cartesian closed category on n generating objects, and that $\mathsf{PPC}(p_1, p_2, ..., p_n)$ is the free cartesian closed poset on n generating elements. We have an obvious CCC map

$$\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)} \longrightarrow \mathsf{PPC}(p_1, p_2, ..., p_n)$$

taking generators to generators, and it extends along the quotient map to $|\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}|$ 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 $|\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}|$, which follows from the Lemma 3.11.1. □

Remark 3.11.4. Corollary 3.11.3 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 algebraic and propositional logics, the fact that there is a generic model (Proposition 3.9.4) allows the general completeness theorem stated in Corollary 3.9.5 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, topological models, and Kripke models. Note that this follows that same pattern that we saw for the "proof irrelevant" case of propostional logic, but in some cases, the proofs require much 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 category $\mathbb{C}$. We regard such a model as "varying over $\mathbb{C}$", just as a presheaf of groups on 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 fact that we use in specializing Proposition 3.9.4 to such variable models is the following, which is one of the fundamental facts of categorical semantics.

Lemma 3.11.5. *For any small category $\mathbb{C}$, the Yoneda embedding*

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

preserves cartesian closed structure.

Proof. We 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 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 $\widehat{\mathbb{C}}(\mathsf{y}X \times \mathsf{y}A, \mathsf{y}B) = \mathsf{y}B^{\mathsf{y}A}(X)$, and we're done. □

Proposition 3.11.6. *For any λ-theory $\mathbb{T}$, we have the following:*

1. *A type A is inhabited,*
$$\mathbb{T} \vdash a : A$$
if, and only if, there is a point
$$1 \to [\![A]\!]$$
in every model $[\![-]\!]$ in a CCC of presheaves $\mathsf{Set}^{\mathbb{C}^{\mathrm{op}}}$ on a small category $\mathbb{C}$.

2. *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 can specialize the general completeness statement of Corollary 3.9.5 to CCCs of the form $\widehat{\mathbb{C}}$ using Lemma 3.11.5, together with the fact that the Yoneda embedding is full (and therefore reflects inhabitation) and faithful (and therefore reflects equations). □

Topological models

See [?]

Kripke models

See [?]

Models based on computability and continuity

See [?]

3.12 Modal operators and monads

See [?]