Lecture Notes:
Introduction to Categorical Logic

[DRAFT: March 12, 2008]

Steven Awodey Andrej Bauer

March 12, 2008

Contents

Foreword

These lecture notes were written in the Fall of 2002 for two introductory courses in categorical logic—the first author gave such a course at the Department of Philosophy at Carnegie Mellon University, and at the same time the second author gave a very similar course at the Department of Mathematics and Physics at University of Ljubljana. This was the third time such a course was given at Carnegie Mellon University, and so the material was deemed mature enough to be written up as lecture notes.

The course is intended for advanced undergraduate students and graduate students who have some background in category theory. We expect that students who have heard and used basic concepts of category theory in undergraduate courses, such as algebraic topology and abstract algebra, will be able to follow the material. We hope that the extensive review of category theory in Chapter 1 will also allow motivated students who lack category-theoretic background to quickly catch up.

Categorical logic is a broad subject, and so an introductory course must necessarily make certain choices. From a desire to keep the required category-theoretic machinery at a minimum we avoided any use of fibered categories. Throughout, our motto was "types are objects, propositions are subobjects". From a technical point of view this approach leads to problems when one considers dependent types, but from an educational point of view it is far outweighed by the simpler setup and transparency of ideas. In particular, we present the semantics of dependent types in terms of locally cartesian categories, and only comment briefly on problems involving equality of types that arise in this kind of semantics.

If you find any mistakes in the notes, and undoubtedly there are some, please contact us. An updated version of the notes and a list of known mistakes is available at `http://andrej.com/catlog/`.

Steve Awodey and Andrej Bauer

Chapter 1

Review of 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 as

$$f : \mathsf{dom}\, f \to \mathsf{cod}\, f$$

- 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 $1_A : A \to A$ which is a unit for composition:

$$1_A \circ f = f \ , \qquad\qquad g \circ 1_A = g \ ,$$

  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 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 The empty category 0

The empty category has no objects and no arrows.

1.1.2 The unit category 1

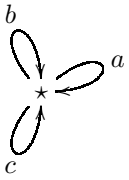
The unit category, also called the terminal category, has one object $\star$ and one arrow $1_\star$:

1.1.3 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:

1.1.4 Groups as categories

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



$a, b, c, \ldots \in G$

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.

1.1.5 Posets as categories

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

$$x \leq x \qquad \text{(reflexive)}$$
$$x \leq y \wedge y \leq z \Longrightarrow x \leq z \qquad \text{(transitive)}$$
$$x \leq y \wedge y \leq z \Longrightarrow x = y \qquad \text{(antisymmetric)}$$

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. 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 posets of open sets $\mathcal{O}X$ of a topological space X, ordered by inclusion.

1.1.6 Sets as categories

Any set S is a category whose objects are the elements of S and the only arrows are the identity arrows. A category in which the only arrows are the identity arrows is a *discrete category*.

1.1.7 Structures as categories

In general structures like groups, topological spaces, posets, etc., determine categories in which 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]

[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.

- Graph is the category of (directed) graphs an graph homomorphisms.
- Poset is the category of posets and monotone maps.

Such categories of structures are generally *large*.

1.1.8 Further Definitions

We recall some further basic notions in category theory.

Definition 1.1.3 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 *full subcategory* $\mathcal{C}'$ of $\mathcal{C}$ is a subcategory of $\mathcal{C}$ such that, for all $A, B \in \mathcal{C}'_0$, if $f : A \to B$ is in $\mathcal{C}_1$ then it is also in $\mathcal{C}'_1$.

Definition 1.1.4 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 an *iso*. If there exists a pair of 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 1.1.5 A *monomorphism*, or *mono*, is a morphism $f : A \to B$ that can be canceled on the left: for all $g : C \to A$, $h : C \to A$,

$$f \circ g = f \circ h \Longrightarrow g = h \ .$$

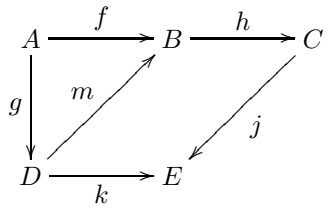
An *epimorphism*, or *epi*, is a morphism $f : A \to B$ that can be canceled on the right: for all $g : B \to C$, $h : B \to A$,

$$g \circ f = h \circ f \Longrightarrow g = h \ .$$

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. (See examples in the next section.)

A more realistic example of morphisms that are both epi and mono but are not iso occurs in the category Top of topological spaces and continuous map because 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:



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:

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

From these we can derive $k \circ g = h \circ h \circ f$.

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 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 and locally small.

Definition 1.2.2 A functor $F : \mathcal{C} \to \mathcal{D}$ is *faithful* when it is injective on morphisms: 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 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, Graphs, 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 $f : G \to H$.

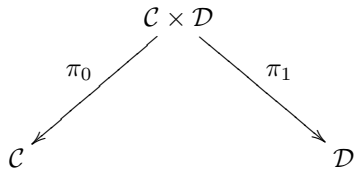
There are also forgetful functors that forget only part of the structure, for example the forgetful functor $U : \mathsf{Ring} \to \mathsf{Group}$ which maps a ring $(R, +, \times)$ to the additive group $(R, +)$ and a ring homomorphism $f : (R, +_R, \cdot_S) \to (S, +_S, \cdot_S)$ to the group homomorphism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

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



which act as indicated in the following diagrams:

Exercise 1.3.1 Show that, for any categories $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$,

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

What does $\cong$ mean here?

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,

$$\begin{array}{c} B \\ \downarrow{\scriptstyle f} \\ A \end{array} \tag{1.1}$$

and as morphisms commutative diagrams over A,

$$\begin{array}{ccc} B & \xrightarrow{\quad g \quad} & B' \\ & {\scriptstyle f}\searrow \quad \swarrow{\scriptstyle f'} & \\ & A & \end{array} \tag{1.2}$$

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

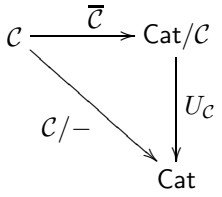
$$\begin{array}{ccc} B & \xrightarrow{\quad g \quad} & B' \\ & {\scriptstyle h\circ f}\searrow \quad \swarrow{\scriptstyle h\circ f'} & \\ & A' & \end{array}$$

The construction of slice categories itself is a functor

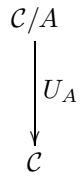
$$\mathcal{C}/- : \mathcal{C} \to \mathsf{Cat}$$

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

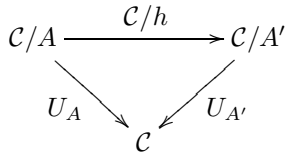
Since Cat is a category, we may form the slice category $\mathsf{Cat}/\mathcal{C}$ for any small category $\mathcal{C}$. The slice functor $\mathcal{C}/-$ factors through the forgetful functor $U_{\mathcal{C}} : \mathsf{Cat}/\mathcal{C} \to \mathsf{Cat}$ via a functor $\overline{\mathcal{C}} : \mathcal{C}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



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



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



1.3.3 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 $\mathcal{C}$. Composition and identity arrows in $\mathcal{C}^{\mathsf{op}}$ are the same as in $\mathcal{C}$. Clearly, the opposite of the opposite of a category is 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 $F^{\mathsf{op}} : \mathcal{C}^{\mathsf{op}} \to \mathcal{D}^{\mathsf{op}}$ such that $-^{\mathsf{op}}$ itself becomes a functor? On what category is it a functor?

1.3.4 Representable functors

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

or $\mathcal{C}(A, B)$. For every $A \in \mathcal{C}$ there is a functor

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

defined by

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

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

$$A \xrightarrow{\;f\;} B \xrightarrow{\;g\;} C \xrightarrow{\;h\;} D \tag{1.3}$$

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 (1.3) 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*.

To summarize, 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

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

1.3.5 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 requirements for a *left group action*:

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

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 $f : A \to B$, $\eta_B \circ Ff = Gf \circ \eta_A$, i.e., the following diagram commutes:

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

As an 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:

$$\begin{array}{ccc} \star & \xrightarrow{b} & \star \\ {\scriptstyle fa}\downarrow & & \downarrow{\scriptstyle ga} \\ \star & \xrightarrow[b]{} & \star \end{array}$$

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 function composition. 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:

$$\mathcal{C} \xrightarrow{F} \mathcal{D} \;\overset{G}{\underset{H}{\Downarrow \theta}}\; \mathcal{E}$$

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

$$\mathcal{C} \;\overset{G}{\underset{H}{\Downarrow \theta}}\; \mathcal{D} \xrightarrow{F} \mathcal{E}$$

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

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 $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$,

$$\mathsf{Cat}(\mathcal{A} \times \mathcal{B}, \mathcal{C}) \cong \mathsf{Cat}(\mathcal{A}, \mathcal{C}^{\mathcal{B}}) \; . \tag{1.4}$$

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

$$\begin{array}{ccc} (GA)B & \xrightarrow{(Gf)_B} & (GA')B \\ {\scriptstyle (GA)g}\downarrow & & \downarrow{\scriptstyle (GA')g} \\ (GA)B' & \xrightarrow[(Gf)_{B'}]{} & (GA')B' \end{array}$$

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":

$$E \underset{t}{\overset{s}{\rightrightarrows}} V$$

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 graph G is a functor $G : (\cdot \rightrightarrows \cdot) \to \mathsf{Set}$, defined by

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

If category theory is worth anything, it *should* be the case that 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:

$$\begin{array}{ccc} G_E & \xrightarrow{\phi_E} & H_E \\ {\scriptstyle \mathsf{src}_G}\downarrow & & \downarrow{\scriptstyle \mathsf{src}_H} \\ G_V & \xrightarrow[\phi_V]{} & H_V \end{array} \qquad\qquad \begin{array}{ccc} G_E & \xrightarrow{\phi_E} & H_E \\ {\scriptstyle \mathsf{trg}_G}\downarrow & & \downarrow{\scriptstyle \mathsf{trg}_H} \\ G_V & \xrightarrow[\phi_V]{} & H_V \end{array}$$

This is precisely the requirement that $e : a \to b$ implies $\phi_E e : \phi_V a \to \phi_V b$.

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-set functor

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

By transposing it 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 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 is called the *Yoneda embedding.*

Theorem 1.4.3 (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 Yoneda Lemma.

Lemma 1.4.4 (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.

Proof. The desired natural isomorphism θ_A maps a natural transformation $\eta \in \mathsf{Nat}(\mathsf{y}A, F)$ to $\eta_A \mathsf{1}_A$. 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

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

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

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

and also

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

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

$$\begin{array}{ccc} \mathcal{C}(A, A) & \xrightarrow{\eta_A} & FA \\ {\scriptstyle \mathcal{C}(f, A)}\downarrow & & \downarrow{\scriptstyle Ff} \\ \mathcal{C}(B, A) & \xrightarrow[\eta_B]{} & FB \end{array}$$

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

$$\begin{array}{ccc} \mathsf{Nat}(\mathsf{y}A, F) & \xrightarrow{\theta_A} & FA \\ {\scriptstyle \mathsf{Nat}(\mathsf{y}g, F)}\uparrow & & \uparrow{\scriptstyle Fg} \\ \mathsf{Nat}(\mathsf{y}A', F) & \xrightarrow[\theta_{A'}]{} & FA' \end{array}$$

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

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

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

Proof. [Proof of Theorem 1.4.3] 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 Yoneda Lemma applied to the case $F = \mathsf{y}B$. Indeed, with notation as in the proof of 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 1.4.5 *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 1.4.6 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 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

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

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 required.

Definition 1.4.7 An *equivalence of categories* is a pair of functors

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

such that

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

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 at interest from a categorical point of view, such as identity of objects.

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

Proposition 1.4.8 *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, which means 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

$$F(Gf) = {\eta_C}^{-1} \circ f \circ \eta_B \ . \tag{1.5}$$

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, (1.5) ensures that η is a natural isomorphism $F \circ G \Longrightarrow 1_{\mathcal{D}}$.

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

Example 1.4.9 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

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

[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

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

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 $\bot_A$ as a special "undefined value". Let us look at the composition of F and G on objects:

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

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

Example 1.4.10 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 1.4.8, q is an equivalence functor. Trivially, it is faithful and surjective on objects. It is also full because $a \leq b$ in P implies $qa \sqsubseteq qb$ in $P/\!\cong$.

1.5 Adjoint Functors

The notion of adjunction is arguably the most important concept unveiled by category theory. It is a general logical and mathematical concept that occurs everywhere and often marks an important and interesting connection between two objects of interest. In logic, adjoint functors are pervasive, although this is only recognizable from the category-theoretic approach to logic.

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 two monotone maps between them,

$$P \underset{g}{\overset{f}{\rightleftarrows}} Q$$

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

$$fx \leq y \iff x \leq gy\,. \tag{1.6}$$

Note that adjointness is *not* a symmetric relation. The map f is the *left adjoint* and g is the *right adjoint*.[4]

Equivalence (1.6) is more conveniently displayed as

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

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 whose only logical operations are conjunction $\wedge$ and implication $\Rightarrow$.[5] 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 $\wedge$ and $\Rightarrow$. The logical rules are given in natural deduction style:

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

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

For example, we read the last two inference rules as "from $A \Rightarrow B$ and A we infer B" and "if from assumption A we infer B, then (without any assumptions)

[4] Remember it like this: the left adjoint stands on the *left* side of $\leq$, the right adjoint stands on the *right* side of $\leq$.

[5] Nothing changes if we consider a calculus with more connectives.

we infer $A \Rightarrow B$", respectively. We indicate assumptions by enclosing them in brackets. The symbol u in $[u : A]$ is a label for the assumption. When an assumption is *discharged* its label is written to the right of the inference 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 prove 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

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

The maps f and g are functors because 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:

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

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 equivalence:

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

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:

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

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 a monotone map, and so is the interior $\mathsf{int} : \mathcal{P}X \to \mathcal{O}X$:

$$\mathcal{O}X \underset{\mathsf{int}}{\overset{i}{\rightleftarrows}} \mathcal{P}X$$

For $U \in \mathcal{O}X$ and $S \in \mathcal{P}X$ we have

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

Therefore, *topological interior is a 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 to the general situation

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

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

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

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

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

Definition 1.5.1 An *adjunction* $F \dashv G$ between functors

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

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

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

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:

$$\begin{array}{ccc} \mathcal{D}(FA, B) & \xrightarrow{\theta_{A,B}} & \mathcal{D}(A, GB) \\ \Big\downarrow{\scriptstyle \mathcal{D}(Ff, g)} & & \Big\downarrow{\scriptstyle \mathcal{C}(f, Gg)} \\ \mathcal{D}(FA', B') & \xrightarrow[\theta_{A',B'}]{} & \mathcal{C}(A', GB') \end{array}$$

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 a *left adjoint* and G is a *right adjoint*.

We have already seen examples of adjoint functors. For any category $\mathcal{B}$ we have functors $- \times \mathcal{B}$ and $-^{\mathcal{B}}$ from Cat to Cat. Recall the isomorphism (1.4),

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

This isomorphism is in fact natural so that

$$- \times \mathcal{B} \dashv -^{\mathcal{B}} \ .$$

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 $f^B = f \circ -$. Then we have, for all $A, C \in \mathsf{Set}$, a natural isomorphism

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

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 (1.6) of adjoint monotone maps between preorders is a special case of Definition 1.5.1.

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 finite paths

$$v_1 \xrightarrow{e_1} v_2 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_{n+1}$$

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 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 functors. If $F \dashv G$, $F \dashv G'$ and $F' \dashv G$ then $F \cong F'$ and $G \cong G'$.*

Proof. Suppose $F \dashv G$ and $F \dashv G'$. By Yoneda Embedding, $GB \cong G'B$ if, and only if, $\mathcal{C}(-, GB) \cong \mathcal{C}(-, G'B)$, which holds because, for any $A \in \mathcal{C}$,

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

Therefore, $G \cong G'$. That $F \dashv G$ and $F' \dashv G$ implies $F \cong F'$ is proved similarly, except that the Yoneda Embedding must be replaced by its covariant version. ∎

1.5.3 The Unit of an Adjunction

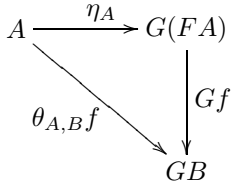
Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be adjoint functors, $F \dashv G$, and let $\theta : \mathcal{D}(F-, -) \to \mathcal{C}(-, G-)$ be the natural isomorphism witnessing the adjunction. For any object $A \in \mathcal{C}$ there is a distinguished morphism $\eta_A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

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

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

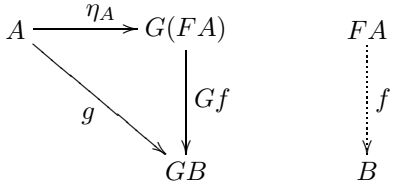
$$\theta_{A,B} f = \theta_{A,B}(f \circ 1_{FA}) = Gf \circ \theta_{A,FA}(1_{FA}) = Gf \circ \eta_A \ ,$$

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



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 $g = Gf \circ \eta_A$:



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,

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

The monoid operation is concatenation of sequences

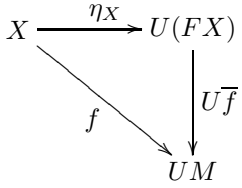
$$x_1 \dots x_m \cdot y_1 \dots y_n = x_1 \dots x_m y_1 \dots y_n \ ,$$

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 free monoid FX is "free" in the sense that for every every monoid M and a function $f : X \to UM$ there exists a unique homomorphism $\overline{f} : FX \to M$ such that the following diagram commutes:



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 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, and let $\theta : \mathcal{D}(F-,-) \to \mathcal{C}(-,G-)$ be the natural isomorphism witnessing the adjunction. For any object $B \in \mathcal{D}$ there is a distinguished morphism $\varepsilon_B = \theta^{-1}_{GB,B} 1_{GB} : F(GB) \to B$,

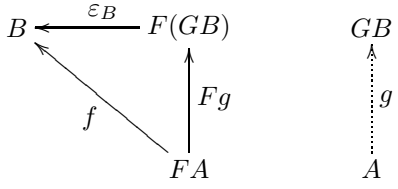
$$\frac{1_{GB} : GB \to GB}{\varepsilon_B : F(GB) \to B}$$

The transformation $\varepsilon : F \circ G \Longrightarrow 1_{\mathcal{D}}$ is natural and 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 counit, which are easily obtained by "turning around" all 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 $f = \varepsilon_B \circ Fg$:



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

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

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

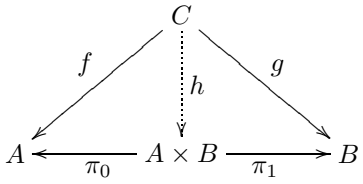
$$\overline{f}x = fx \ ,$$

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

1.6 Limits and Colimits

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 all 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:



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

$$\forall\, C{:}\mathcal{C}\,.\,\forall\, f{:}C \to A\,.\,\forall\, g{:}C \to B\,.\,\exists!\, h{:}C \to A \times B\,.\\ (\pi_0 \circ h = f \wedge \pi_1 \circ h = g)$$

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 exists 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[6] 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 the axiom of choice. When we view binary products as an operation, we say that "$\mathcal{C}$ has chosen products". The same holds for other specific and general 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 object

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

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 $\mathbf{1}$ consisting of one object and one morphism.

Exercise 1.6.1 Prove that if 1 and $1'$ 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.[7] Does Field have a terminal object?

1.6.3 Equalizers

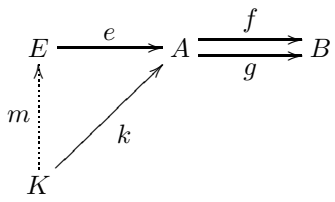
Given objects and morphisms

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

[6] More precisely, binary product is a functor from $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$, cf. Section 1.6.11.

[7] 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.

we say that e *equalizes* f and g when $f \circ e = g \circ e$.[8] 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 $m : K \to E$ such that $k = e \circ m$:



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 a single 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.[9]

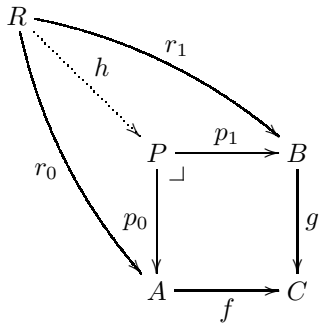
1.6.4 Pullbacks

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 $r_0 : R \to A$, $r_1 : R \to B$ are such that $f \circ r_0 = g \circ r_1$, then there exists a unique $h : R \to P$

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

[9] 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 $f[U] = V \cap \mathsf{im}(f)$.

such that $r_0 = p_0 \circ h$ and $r_1 = p_1 \circ h$:



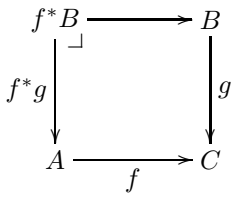
We indicate that P is a pullback by drawing a square corner next to it, as in the above diagram. Sometimes we denote the pullback P by $A \times_C B$.

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

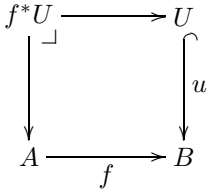
$$P = \big\{ \langle x, y \rangle \in A \times B \;\big|\; fx = gy \big\}$$

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

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

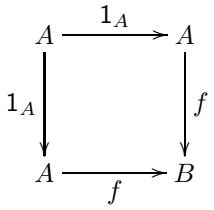


We think of $f^*g : f^*B \to A$ as the inverse image of B along f. This terminology is explained by looking at the pullback of a subset inclusion $u : U \hookrightarrow C$ along a function $f : A \to C$ in the category Set:



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

Exercise 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:

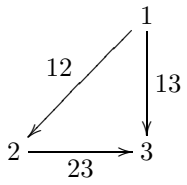


1.6.5 Limits

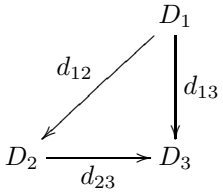
Let us now define a general notion of a limit.

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



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



(1.7)

Given an object $A \in \mathcal{C}$, there is a *constant diagram* of shape $\mathcal{I}$, which is the constant functor $\Delta_A : \mathcal{I} \to \mathcal{C}$ that maps every object to A and every morphism to 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).

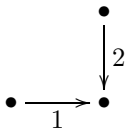
A *limit* of a diagram $D : \mathcal{I} \to \mathcal{C}$ is a terminal object in $\mathsf{Cone}(D)$. Explicitly, a limit of D is given by a cone (L, λ) such that for every other cone (A, α) there exists a *unique* morphism $f : A \to L$ such that $\alpha_i = \lambda_i \circ f$ for all $i \in \mathcal{I}$.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



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

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

$$\left\{ A_i \in \mathcal{C} \;\middle|\; i \in I \right\} \ .$$

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

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

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

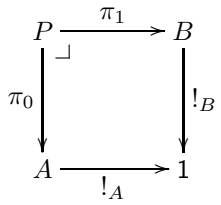
Exercise 1.6.6 Prove that a limit, when it exists, is unique up to isomorphism.

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



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

Proposition 1.6.8 *The following are equivalent for a category $\mathcal{C}$:*

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

$$E \xrightarrow{\ e\ } P \underset{g}{\overset{f}{\rightrightarrows}} Q$$

For every $i \in \mathcal{I}$ there is a morphism $\varepsilon_i : E \to D_i$, namely $\varepsilon_i = \pi_i^P \circ e$. 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 $\mathcal{I}$,

$$Du \circ \varepsilon_i = Du \circ \pi_i^P \circ e = \pi_u^Q \circ f \circ e = \pi_u^Q \circ g \circ e = \pi_j^P \circ e = \varepsilon_j \ .$$

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

$$\pi_u^Q \circ g \circ t = \pi_j^P \circ t = t_j = Du \circ t_i = Du \circ \pi_i^P \circ t = \pi_u^Q \circ f \circ t \ ,$$

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 $i \in \mathcal{I}$

$$\varepsilon_i \circ k = \pi_i^P \circ e \circ k = \pi_i^P \circ t = \alpha_i$$

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 $\pi_i^P \circ e \circ m = \pi_i^P \circ e \circ k$ for all $i \in \mathcal{I}$. This last equality holds because

$$\pi_i^P \circ e \circ k = \pi_i^P \circ t = \alpha_i = \varepsilon_i \circ m = \pi_i^P \circ e \circ m \ .$$

■

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

Limits of presheaves

Let $\mathcal{C}$ be a locally small category. Then the presheaf category $\widehat{\mathcal{C}} = \mathsf{Set}^{\mathcal{C}^{\mathrm{op}}}$ 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:

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

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 category $\mathcal{C}^{\mathsf{op}}$:

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

Equivalently, 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 $i \in \mathcal{I}$.

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

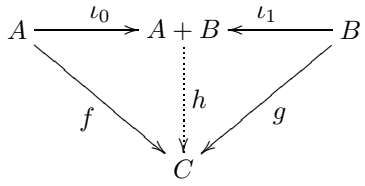
$$\operatorname{colim} D \qquad \operatorname{colim}_{i\in\mathcal{I}} D_i \qquad \varinjlim_{i\in\mathcal{I}} D_i \ .$$

Colimits are also called *inductive limits*.

Exercise 1.6.9 Formulate the dual of Proposition 1.6.7 and Proposition 1.6.8 for colimits (coequalizers are defined in Subsection 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:



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

The coproduct $A + B$ is the colimit of the diagram $D : 2 \to \mathcal{C}$, 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

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

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 Suppose A and B are Abelian groups.[10] What is the difference between their coproduct in the category Group of groups, and their coproduct in the category AbGroup of Abelian groups?

1.6.8 The initial object

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.

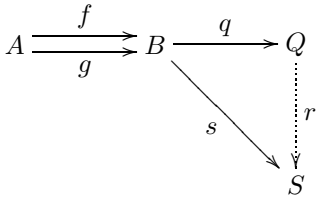
Exercise 1.6.12 Show that in the category of Abelian groups finite products and coproducts agree, that is $\mathsf{0} \cong \mathsf{1}$ and $A \times B \cong A + B$.

1.6.9 Coequalizers

Given objects and morphisms

$$A \underset{g}{\overset{f}{\rightrightarrows}} B \xrightarrow{\ q\ } Q$$

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 $s = r \circ q$:



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

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

$$fx = gy \Longrightarrow x \sim y \ .$$

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, coequalizers can be thought of as quotients of those equivalence relations that that can be defined (generated) by a single equation.

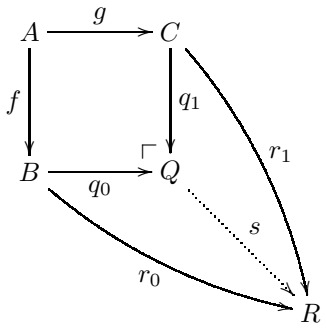
Exercise 1.6.13 Show that a coequalizer is an epimorphism, i.e., if $q : B \to Q$ is a coequalizer of f and g, then, for all $u, v : Q \to T$, $u \circ q = v \circ q$ implies $u = v$. [Hint: use the duality between limits and colimits and Exercise 1.6.3.]

Definition 1.6.14 A morphism is a *regular epi* if it is a coequalizer.

The difference between epis and regular epis is best 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.[11]

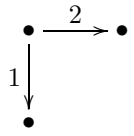
1.6.10 Pushouts

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



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 denotes by $B +_A C$.

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



[11] 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.

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 and Colimits as Adjoints

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 and the only morphism $\mathsf{1}_\star$ to $\mathsf{1}_A$.

Now if $\mathcal{C}$ has a terminal object $\mathsf{1}_\mathcal{C}$ we can ask whether the corresponding functor $\mathsf{1}_\mathcal{C} : \mathsf{1} \to \mathcal{C}$ has any adjoints. 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 $\star$. This functor is indeed adjoint to $\mathsf{1}_\mathcal{C}$ because, for every $A \in \mathcal{C}$ we have a (trivially natural) bijective correspondence

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

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

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

If $\mathcal{C}$ has binary products then they can be viewed as 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 $\pi_1 \circ (f \times g) = g \circ \pi_1$,

$$\begin{array}{ccccc} A & \xleftarrow{\pi_0} & A \times B & \xrightarrow{\pi_1} & B \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f \times g} & & \downarrow{\scriptstyle g} \\ A' & \xleftarrow[\pi_0]{} & A' \times B' & \xrightarrow[\pi_1]{} & B' \end{array}$$

The binary product functor has a left adjoint, namely the diagonal diagram functor

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

defined by $\Delta A = \langle A, A\rangle$, $\Delta f = \langle f, f\rangle$. 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 left adjoint to the diagonal functor:

$$(- + -) \dashv \Delta \dashv (- \times -) \,.$$

In general, suppose $\mathcal{C}$ has limits of shape $\mathcal{I}$. Then the limit construction is a functor

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

that maps each diagram $D \in \mathcal{C}^{\mathcal{I}}$ to its limit $\lim D$. In the opposite direction there is the constant diagram functor

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

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

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

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

$$\mathrm{colim} \dashv \Delta \dashv \lim \,.$$

Exercise 1.6.15 How are the functors $\lim : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$, $\mathrm{colim} : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$, and $\Delta : \mathcal{C} \to \mathcal{C}^{\mathcal{I}}$ defined on morphisms?

1.6.12 Preservation of Limits and Colimits by Functors

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

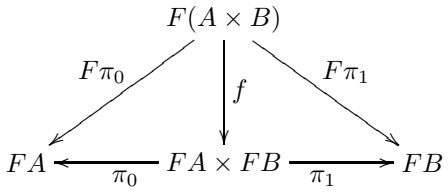
$$A \xleftarrow{\pi_0} A \times B \xrightarrow{\pi_1} B$$

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: [12]



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 1.6.16 *(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 1.6.17 *For a locally small category $\mathcal{C}$, the Yoneda embedding $\mathsf{y} : \mathcal{C} \to \widehat{\mathcal{C}}$ preserves all limits that exist in $\mathcal{C}$.*

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

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

[12] 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.

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

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

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

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

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

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

$$\mathcal{C}(\operatorname*{colim}_{i \in \mathcal{I}} D_i, A) \cong \lim_{i \in \mathcal{I}} \mathcal{C}(D_i, A) \ .$$

Exercise 1.6.18 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 1.6.19 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 1.6.20 *Right adjoints preserve limits, and left adjoints preserve colimits.*

Proof. Suppose we have adjoint functors

$$\mathcal{C} \underset{G}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathcal{D}$$

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 $A \in \mathcal{C}$,

$$\mathcal{C}(A, G(\lim D)) \cong \mathcal{D}(FA, \lim D) \cong \lim_{i \in \mathcal{I}} \mathcal{D}(FA, D_i) \cong \lim_{i \in \mathcal{I}} \mathcal{C}(A, GD_i) \cong \mathcal{C}(A, \lim(G \circ D)) \ ,$$

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

Type Theories

2.1 Algebraic Theories

In this section we study a general approach to algebraic structures such as groups, rings, modules, and lattices. These are characterized by axiomatizations which involve only constants, operations, and equations. It is important that the operations are defined everywhere, which excludes two important examples: fields because the inverse of 0 is undefined, and categories because composition is defined only for some pairs of morphisms.

Let us start with the quintessential algebraic theory—the theory of a group. A group is a set G with a binary operation $\cdot : G \times G \to G$, satisfying the two axioms

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

We want to pay attention to the logical form of the axioms, because axioms with a simpler logical structure can be interpreted more widely. The second axiom, which expresses the existence of a unit and inverse elements, is particularly unsatisfactory because it involves nested quantifiers.

If we require the unit to be a distinguished *constant* $e \in G$ and the inverse to be an *operation* $^{-1} : G \to G$ we obtain an equivalent formulation in which all axioms are *equations*:

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

It is understood that equality is an equivalence relation and that we may substitute equals for equals. Notice that the universal quantifier is not needed anymore, provided we interpret the variables as ranging freely over G. In fact, we do not need to explicitly mention the underlying set G at all. Additionally, a

constant can be thought of as a nullary operation, i.e., a function $1 \to G$. This leads to the general definition of an algebraic theory.

Definition 2.1.1 A *signature* Σ for an algebraic theory consists of a family of sets $\{\Sigma_k\}_{k\in\mathbb{N}}$. The elements of Σ_k are called the *k-ary operations*. In particular, the elements of Σ_0 are the *nullary operations* or *constants*.

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

1. variables x, y, z, ..., are terms,
2. if $\langle t_1, \ldots, t_k\rangle$ is a k-tuple of terms and $f \in \Sigma_k$ is a k-ary operation then $f\langle t_1, \ldots, t_k\rangle$ is a term.

Definition 2.1.2 (cf. Definition 2.1.14) An *algebraic theory* $\mathbb{A} = (\Sigma, A)$ is given by a signature Σ and a set A of *axioms*, which are equations between terms.

Algebraic theories are also called *equational theories* and *Lawvere theories*.

Example 2.1.3 The theory of a commutative ring with unit is an algebraic theory. There are two nullary operations (constants) 0 and 1, a unary operation $-$, and two binary operations $+$ and $\cdot$. The equations are:

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

Example 2.1.4 The theory with no operations and no equations is the theory of a set.

Example 2.1.5 The theory with one constant and no equations is the theory of a *pointed set*, cf. Example 1.4.9.

Example 2.1.6 Let R be a ring. A left R-module is an algebraic theory. It has one constant 0, a unary operation $-$, a binary operation $+$, and for each $a \in R$ a unary operation $\overline{a}$, called *scalar multiplication by a*. The following equations hold:

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

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

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

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

Example 2.1.7 In computer science, inductive datatypes are examples of algebraic theories. For example, the datatype of binary trees with leaves labeled by integers might be defined as follows in a programming language:

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

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

2.1.1 Many-sorted algebraic theories

It is sometimes necessary to consider algebraic theories with more than one set. In Example 2.1.6 we saw that the theory of a left R-module is algebraic, for a fixed ring R. However, if we wanted to have a general theory of left modules, we would need an algebraic structure consisting of *two* carrier sets, a ring R and a module M, together with operations and equations. To cover such examples we would have to consider *many-sorted algebraic theories*. Here we only give several motivating examples of many-sorted algebraic theories, and postpone the general study of many-sorted theories to subsequent sections.

Example 2.1.8 As already mentioned, the theory of left modules is a two-sorted algebraic theory. There are two sorts, R and M. The axioms express the fact that R is a ring, M is a commutative group, and that scalar multiplication has the desired properties, cf. Example 2.1.6.

Example 2.1.9 The theory of a directed graph is a two-sorted algebraic theory with a sort E for edges and a sort V for vertices. There are two unary operations $\mathtt{src} : E \to V$ and $\mathtt{trg} : E \to V$, which give the source and the target of an edge. There are no equations.

A *symmetric graph* is a graph in which for every edge $e : a \to b$ there is an edge $\overline{e} : b \to a$ in the other direction. This can be axiomatized with a unary operation $e \mapsto \overline{e}$ which satisfies the axioms

$$\mathsf{src}\,\overline{e} = \mathsf{trg}\,e\ , \qquad\qquad \mathsf{trg}\,\overline{e} = \mathsf{src}\,e\ .$$

A *simple graph* is one in which every two vertices are connected with at most one edge. An axiom which expresses this fact is

$$(\mathsf{src}\,e = \mathsf{src}\,f \wedge \mathsf{trg}\,e = \mathsf{trg}\,f) \Longrightarrow e = f\ .$$

However, this axiom has the form of an implication, so it seems that the theory of a simple graph is not algebraic.

Example 2.1.10 Modern computers are equipped with *random access memory (RAM)*. An idealized RAM consists of a number of *memory locations* indexed by a set A of *addresses*. Each memory location contains *data* which is an element of a set D. For example, the addresses might be 64-bit integers and data might be 8-bit integers.[1] The two basic operations on memory are a *memory lookup* and a *memory update*.

Let M be the set of all possible memory configurations. For each address $a \in A$ we have an operation $\mathtt{lookup}_a : M \to D$ which returns the content of location a, and an update operation $\mathtt{update}_a : M \times D \to M$ which updates the content of location a and returns the updated memory configuration. These operations satisfy the following equations, for all $m \in M$, $d, e \in D$, and $a, b \in A$ with $a \neq b$:

$$\begin{aligned}
\mathtt{lookup}_a\,(\mathtt{update}_a\,\langle m, d\rangle) &= d \\
\mathtt{lookup}_a\,(\mathtt{update}_b\,\langle m, d\rangle) &= \mathtt{lookup}_a\,m \\
\mathtt{update}_a\,\langle \mathtt{update}_b\,\langle m, e\rangle, d\rangle &= \mathtt{update}_b\,\langle \mathtt{update}_a\,\langle m, d\rangle, e\rangle \\
\mathtt{update}_a\,\langle \mathtt{update}_a\,\langle m, d\rangle, e\rangle &= \mathtt{update}_a\,\langle m, e\rangle
\end{aligned}$$

We see that random access memory can be formalized as a *two-sorted algebraic theory* with a sort M of memory configurations and a sort D of data. The addresses A are *not* a sort, they just parameterize the lookup and update operations. Had we made A into a sort, we would encounter problems because the second and the third axioms above would have to express the condition $a \neq b$, which cannot be done with equations in general.

2.1.2 Models of Algebraic Theories

Let us now consider what a *model* of an algebraic theory is. In classical algebra, a group is given by a set G, an element $e \in G$, a function $m : G \times G \to G$ and a function $i : G \to G$, satisfying the group axioms:

$$\begin{aligned}
m\langle x, m\langle y, z\rangle\rangle &= m\langle m\langle x, y\rangle, z\rangle\ , \\
m\langle x, i\,x\rangle &= m\langle i\,x, x\rangle = e\ , \\
m\langle x, e\rangle &= m\langle e, x\rangle = x\ .
\end{aligned}$$

We would like to generalize this notion so that we can speak of models of group theory in categories other than Set. This can be accomplished by translating everything into the language of category theory: a group is given by an object $G \in \mathsf{Set}$ and three morphisms

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

[1] An n-bit integer is an integer between 0 and $2^n - 1$.

Associativity of m is expressed by commutativity of the following diagram:

$$\begin{array}{ccc} G \times G \times G & \xrightarrow{m \times \pi_2} & G \times G \\ {\scriptstyle \pi_0 \times m}\downarrow & & \downarrow{\scriptstyle m} \\ G \times G & \xrightarrow[m]{} & G \end{array}$$

Similarly, the axioms for the unit and the inverse are expressed by commutativity of the following diagrams:

$$\begin{array}{ccccc} G \times 1 & \xrightarrow{1_G \times e} & G \times G & \xleftarrow{e \times 1_G} & 1 \times G \\ & {\scriptstyle \pi_0}\searrow & \downarrow{\scriptstyle m} & \swarrow{\scriptstyle \pi_1} & \\ & & G & & \end{array} \qquad \begin{array}{ccccc} G & \xrightarrow{\langle 1_G, i\rangle} & G \times G & \xleftarrow{\langle i, 1_G\rangle} & G \\ {\scriptstyle !_G}\downarrow & & \downarrow{\scriptstyle m} & & \downarrow{\scriptstyle !_G} \\ 1 & \xrightarrow[e]{} & G & \xleftarrow[e]{} & 1 \end{array}$$

We see that this formulation makes sense in any category $\mathcal{C}$ with finite products.

In general, an *interpretation* I of the terms of a theory $\mathbb{A}$ in a category $\mathcal{C}$ is given by an object $I\mathbb{A} \in \mathcal{C}$ and, for each basic operation f of arity k, a morphism $If : (I\mathbb{A})^k \to I\mathbb{A}$. In particular, basic constants are interpreted as morphisms $1 \to I\mathbb{A}$. A general term t is always interpreted together with a *context* of variables $x_1, \ldots, x_n$, where the variables appearing in t must be among the variables appearing in the context. We write

$$x_1, \ldots, x_n \mid t \tag{2.1}$$

to indicate that the term t is to be understood in context $x_1, \ldots x_n$. The interpretation of (2.1) is a morphism $It : (I\mathbb{A})^n \to I\mathbb{A}$, determined by the following rules:

1. The interpretation of a variable x_i is the i-th projection $\pi_i : (I\mathbb{A})^n \to I\mathbb{A}$.
2. A term of the form $f\langle t_1, \ldots, t_k\rangle$ is interpreted as the composition

$$(I\mathbb{A})^n \xrightarrow{\langle It_1, \ldots, It_k\rangle} (I\mathbb{A})^k \xrightarrow{\qquad If \qquad} \mathbb{A}$$

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

It is clear from this that the interpretation of a term really depends on the context. For example, the term $f\, x_1$ is interpreted as a morphism $If : I\mathbb{A} \to I\mathbb{A}$ in context x_1, and as the morphism $If \circ \pi_1 : (I\mathbb{A})^2 \to I\mathbb{A}$ in the context x_1, x_2.

Suppose u and v are terms in context $x_1, \ldots, x_n$. Then we say that the equation $u = v$ is *satisfied* by the interpretation I if Iu and Iv are interpreted as the same morphism. In particular, suppose $u = v$ is an axiom of the theory, and let $x_1, \ldots, x_n$ be all the variables appearing in u and v. We say that I *satisfies the axiom* $u = v$ when $x_1, \ldots, x_n \mid u$ and $x_1, \ldots, x_n \mid u$ are interpreted as the same morphism by I.

Definition 2.1.11 (cf. Definition 2.1.19) A *model* M of an algebraic theory $\mathbb{A}$ in a category $\mathcal{C}$ with finite products is an interpretation of the theory that satisfies all the axioms of the theory.

Example 2.1.12 A model of the theory of a set in a category $\mathcal{C}$ with finite products is determined by an object $A \in \mathcal{C}$. In other words, the mathematicians who live in category $\mathcal{C}$ think of the objects of the category as ordinary sets.

Exercise 2.1.13 A model of group theory in Set is just an ordinary group. But we can also ask what a model of a group is in an arbitrary category with finite products. Determine what the models of a group are in the following categories: the category of finite sets FinSet, the category of topological spaces Top, the category of graphs Graph, and the category of groups Group.

Hint: Only the last case is tricky. Before thinking about it, prove the following lemma [Bor94b, Lemma 3.11.6]. Let G be a set provided with two binary operations $\cdot$ and $\star$ and a common unit e, so that $x \cdot e = e \cdot x = x \star e = e \star x = x$. Suppose the two operations commute, i.e., $(x \star y) \cdot (z \star w) = (x \cdot z) \star (y \cdot w)$. Then they coincide, are *commutative* and associative.

2.1.3 Algebraic theories as categories

The preceding account of models of algebraic theories is rather syntactic in nature. We would prefer an approach that emphasizes the algebraic point of view. The first step towards this is a *representation-free* notion of algebraic theories.

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

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

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

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

There may be various reasons why we prefer to work with one formulation of group theory rather than another, but this should not be reflected in the general idea of what is a group. We want to avoid particular choices of basic constants, operations, and axioms.[2] This is accomplished by a rather brute method—simply take *all* operations built from unit, multiplication, and inverse as basic, and *all* valid equations of group theory as axioms. We can even go one step further and collect all the operations into a category. We describe the construction of such a category for a general algebraic theory.

[2] This is akin to a basis-free theory of vector spaces: it is better to formulate the idea of a vector space without speaking explicitly of vector bases, even though every vector space has one. Without a doubt, vector bases are important, but they really are a *derived* concept.

Let $\mathbb{A}$ be an algebraic theory. We construct a category, which is also denoted by $\mathbb{A}$, as follows. As objects we take sequences of variables, called *contexts*,

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

A morphism from $[x_1, \ldots, x_m]$ to $[x_1, \ldots, x_n]$ is an n-tuple $\langle t_1, \ldots, t_n \rangle$, where each t_k is a term of the theory whose variables are among $x_1, \ldots, x_m$. Two such morphisms $\langle t_1, \ldots, t_n \rangle$ and $\langle u_1, \ldots, u_n \rangle$ are equal if, and only if, the axioms of the theory imply that $t_k = u_k$ for every $k = 1, \ldots, n$.[3] The composition of morphisms

$$\begin{aligned} \langle t_1, \ldots, t_m \rangle &: [x_1, \ldots, x_k] \to [x_1, \ldots, x_m] \\ \langle u_1, \ldots, u_n \rangle &: [x_1, \ldots, x_m] \to [x_1, \ldots, x_n] \end{aligned}$$

is the morphism $\langle v_1, \ldots, v_n \rangle$ whose i-th component is obtained by simultaneously substituting in u_i the terms $t_1, \ldots, t_m$ for the variables $x_1, \ldots, x_m$:

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

The identity morphism on $[x_1, \ldots, x_n]$ is $\langle x_1, \ldots, x_n \rangle$. Observe that the object $[x_1, \ldots, x_{n+m}]$ is the product of $[x_1, \ldots, x_n]$ and $[x_1, \ldots, x_m]$ so that all finite products exist. Furthermore, every object is a product of finitely many instances of the object $[x_1]$.

The category $\mathbb{A}$ contains precisely the same "algebraic" information as the theory $\mathbb{A}$ which it was built from. Thus we are lead to the following alternative definition.

Definition 2.1.14 (cf. Definition 2.1.2) An *algebraic theory* $\mathbb{A}$ is a small category with finite products whose objects form a sequence $A^0, A^1, A^2, \ldots$ such that $A^m \times A^n = A^{m+n}$ for all $m, n \in \mathbb{N}$. In particular, $1 = A^0$ is the terminal object and every object is a product of finitely many copies of $A = A^1$.

An algebraic theory $\mathbb{A}$ in the sense of the above definition determines an algebraic theory in the sense of Definition 2.1.2 as follows. As basic operations with arity k we take the morphisms $A^k \to A$. There is a canonical interpretation in $\mathbb{A}$ of terms built from variables and morphisms $A^k \to A$, namely each morphism is interpreted by itself. An equation $u = v$ is taken as an axiom of the theory $\mathbb{A}$ if the canonical interpretations of u and v coincide.

The new view of algebraic theories immediately suggest some interesting examples.

Example 2.1.15 The algebraic theory $\mathcal{C}^\infty$ of smooth maps is the category whose objects are n-dimensional Euclidean spaces 1, $\mathbb{R}$, $\mathbb{R}^2$, $\ldots$, and whose

[3] Strictly speaking, morphisms are *equivalence classes* of terms, where two terms are equivalent when the theory proves them to be equal. It is cumbersome to work with equivalence classes of terms, so we prefer to work directly with terms but keep in mind that equality between them is equality in the algebraic theory.

morphisms are $\mathcal{C}^\infty$-maps between them. Recall that a $\mathcal{C}^\infty$-map $f : \mathbb{R}^n \to \mathbb{R}$ is a function which has all higher partial derivatives, and that a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is a $\mathcal{C}^\infty$-map when its compositions $\pi_k \circ f : \mathbb{R}^n \to \mathbb{R}$ with projections $\pi_k : \mathbb{R}^m \to \mathbb{R}$ are $\mathcal{C}^\infty$-maps.

Example 2.1.16 Recall that a *(total) recursive function* $f : \mathbb{N}^m \to \mathbb{N}^n$ is one that can be computed by a Turing machine. This means that there exists a Turing machine which on input $\langle a_1, \ldots, a_m \rangle$ outputs the value of $f\langle a_1, \ldots, a_m \rangle$. The algebraic theory Rec of recursive functions is the category whose objects are finite powers of the natural numbers $1, \mathbb{N}, \mathbb{N}^2, \ldots$, and whose morphisms are recursive functions between them. The reason for considering this theory is that its models in a category $\mathcal{C}$ with finite products give a a theory of computability in $\mathcal{C}$, cf. Example 2.1.25.

Example 2.1.17 In a category $\mathcal{C}$ with finite products every object $A \in \mathcal{C}$ determines a full subcategory consisting of the finite powers $1, A, A^2, A^3, \ldots$ and morphisms between them. This is the theory of the object A.

Exercise 2.1.18 In Example 2.1.4 we saw that the theory $\mathbb{S}$ with no operations and no axioms is the theory of a set. Prove that the corresponding category $\mathbb{S}$ is equivalent to $\mathsf{FinSet}^{\mathsf{op}}$ where FinSet is the category of finite sets and functions.

2.1.4 Models of Algebraic Theories as Functors

Having successfully turned the notion of an algebraic theory into a special kind of category, we naturally want to know what we can do about the notion of a model.

Suppose $\mathbb{A}$ is a theory and M is a model of $\mathbb{A}$ in a category $\mathcal{C}$. Then the interpretation M determines a functor $M : \mathbb{A} \to \mathcal{C}$ from the corresponding category $\mathbb{A}$, defined on objects by

$$M[x_1, \ldots, x_k] = (M\mathbb{A})^k \,,$$

and on morphisms by the following rules:

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

2. The morphism
$$\langle f\langle t_1, \ldots, t_m \rangle \rangle : [x_1, \ldots, x_k] \to [x_1]$$
is mapped to the composition
$$(M\mathbb{A})^m \xrightarrow{\langle Mt_1, \ldots, Mt_m \rangle} (M\mathbb{A})^k \xrightarrow{\qquad Mf \qquad} \mathbb{A}$$
where $Mt_i : (M\mathbb{A})^n \to M\mathbb{A}$ is the value of M on the morphisms $\langle t_i \rangle : [x_1, \ldots, x_k] \to [x_1]$, for $i = 1, \ldots, m$, and Mf is the interpretation of the basic operation f.

3. The morphism

$$\langle t_1, \ldots, t_m \rangle : [x_1, \ldots, x_k] \to [x_1, \ldots, x_m]$$

is mapped to the morphism $\langle Mt_1, \ldots, Mt_m \rangle$ where Mt_i is the value of M on the morphism $\langle t_i \rangle : [x_1, \ldots, t_k] \to [x_1]$.

That $M : \mathbb{A} \to \mathcal{C}$ really is a functor follows from the assumption that the interpretation M is a model, which means that all the equations of the theory are satisfied by it. Observe that the functor M is defined in such a way that it preserves finite products.

Suppose $N : \mathbb{A} \to \mathcal{C}$ is a functor from the category $\mathbb{A}$ that corresponds to the theory $\mathbb{A}$. When does it determine a model of the theory? Clearly, if N is to be a model of a theory, then it must interpret variables as projections, $N(x_i : [x_1, \ldots, x_n] \to [x_1]) = \pi_i$, which only makes sense if $N[x_1, \ldots, x_n] = (N[x_1])^n$. In other words, N must preserve finite products. But that is all that is required because functoriality of N guarantees that *all* valid equations of the theory are satisfied, so the axioms are certainly satisfied as well. Thus we see that models of algebraic theories are just finite products preserving functors.

Definition 2.1.19 (cf. Definition 2.1.11) A *model* of an algebraic theory $\mathbb{A}$ in a category $\mathcal{C}$ with finite products is a functor $M : \mathbb{A} \to \mathcal{C}$ which preserves finite products.

So far we have not spoken of *homomorphisms* between models of an algebraic theory. Now a suitable notion presents itself: since models are functors, homomorphisms between models are natural transformations.

Definition 2.1.20 Let $\mathbb{A}$ be an algebraic theory and let $\mathcal{C}$ be a category with finite products. The *category* $\mathsf{Mod}_{\mathcal{C}}(\mathbb{A})$ *of* $\mathbb{A}$*-models in* $\mathcal{C}$ has as objects functors $M : \mathbb{A} \to \mathcal{C}$ that preserve finite products and as morphisms natural transformations between such functors.

Definition 2.1.21 An *algebraic category* is a category that is equivalent to a category of models $\mathsf{Mod}_{\mathcal{C}}(\mathbb{A})$ of an algebraic theory.

Example 2.1.22 At the beginning of this section we mentioned that the theory of a field is not algebraic because inverse of 0 is undefined. In principle there could be an equivalent algebraic formulation of the theory of a field which would somehow circumvent this problem. We can now show that this is not the case by proving that the category Field of fields and field homomorphisms is not algebraic.

First observe that a category of models $\mathsf{Mod}_{\mathcal{C}}(\mathbb{A})$ has a terminal object because $\mathcal{C}$ has a terminal object $\mathsf{1}$ and the constant functor $\Delta_{\mathsf{1}} : \mathbb{A} \to \mathcal{C}$ which maps every context to $\mathsf{1}$ is a model. The functor Δ_{1} is the terminal object in $\mathsf{Mod}_{\mathcal{C}}(\mathbb{A})$ because it is the terminal functor in the functor category $\mathcal{C}^{\mathbb{A}}$. Now in

order to see that Field is not algebraic it is sufficient to show that there is no terminal field, which was Exercise 1.6.2.

By the way, the solution to Exercise 1.6.2 goes as follows: if T were a terminal field then by considering the unique homomorphism $\mathbb{Z}_2 \to T$ we see that $1+1=0$ in T, and by the unique homomorphism $\mathbb{Z}_3 \to T$ we see that $1+1+1=0$ in T, from which we get the impossibility $1=0$.

Example 2.1.23 Let us see what the preceding definitions give us in the case of group theory $\mathbb{G}$. Recall that the category $\mathbb{G}$ consists of contexts $[x_1, \ldots, x_n]$ and terms built from variables and the basic group operations. A finite product preserving functor $M : \mathbb{G} \to \mathsf{Set}$ is then determined up to natural isomorphism by its action on the context $[x_1]$ and the terms representing the basic operations. If we set

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

then (G, e, i, m) is a just a group with unit e, inverse i and multiplication m. That G satisfies the axioms for groups follows from functoriality of M. Conversely, any group (G, e, i, m) determines a finite product preserving functor $M_G : \mathbb{G} \to \mathsf{Set}$ defined by

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

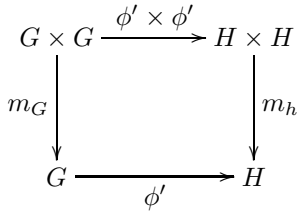
This suggests that $\mathsf{Mod}_{\mathsf{Set}}(\mathbb{G})$ is equivalent to Group, provided both categories have the same notion of morphisms.

Suppose then that (G, e_G, i_G, m_G) and (H, e_H, i_H, m_H) are groups, and let $\phi : M_G \Longrightarrow M_H$ be a natural transformation between the corresponding functors. Then ϕ is already determined by its component at $[x_1]$ because by naturality the following diagram commutes, for $1 \leq k \leq n$:

$$\begin{array}{ccc} G^n & \xrightarrow{\phi_{[x_1,\ldots,x_n]}} & H^n \\ {\scriptstyle G\pi_k = \pi_k}\downarrow & & \downarrow{\scriptstyle H\pi_k = \pi_k} \\ G & \xrightarrow[\phi_{[x_1]}]{} & H \end{array}$$

If we write $\phi' = \phi_{[x_1]}$ then it follows that $\phi_{[x_1,\ldots,x_n]} = \phi' \times \cdots \times \phi'$. Again, by

naturality of ϕ we see that the following diagram commutes:



Similar commutative squares show that ϕ' preserves the unit and commutes with the inverse operation, therefore $\phi' : G \to H$ is indeed a group homomorphism. Conversely, a group homomorphism $\psi' : G \to H$ determines a natural transformation $\psi : G \Longrightarrow H$ whose component at $[x_1, \ldots, x_n]$ is the n-fold product $\psi' \times \cdots \psi' : G^n \to H^n$. This demonstrates that

$$\mathsf{Mod}_{\mathsf{Set}}(\mathbb{G}) \simeq \mathsf{Group}\ .$$

Example 2.1.24 Consider the theory $\mathcal{C}^\infty$ of smooth maps from Example 2.1.15. A model of this theory in Set is a finite product preserving functor $A : \mathcal{C}^\infty \to \mathsf{Set}$. Up to natural isomorphism it can be described as follows. A $\mathcal{C}^\infty$-model is given by a set A and for every smooth map $f : \mathbb{R}^n \to \mathbb{R}$ a function $Af : A^n \to A$ such that if $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^m \to \mathbb{R}$, $i = 1, \ldots, n$, are smooth maps then, for all $a_1, \ldots, a_m \in A$,

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

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

Example 2.1.25 In Example 2.1.16 we defined the algebraic theory Rec with objects the finite powers of $\mathbb{N}$ and morphisms recursive functions. We now consider the category of its set-theoretic models $\mathcal{R} = \mathsf{Mod}_{\mathsf{Set}}(\mathsf{Rec})$.

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

2.1.5 Completeness and Universal Models

In the last section of this chapter we summarize our method of studying algebraic theories. The same method will be used for studying other kinds of theories as

well. We then conclude the chapter by proving that categorical semantics of algebraic theories is complete.

Categorical logic has two sides, the *logical* and the *categorical* one. The logical side is embodied in the notion of a *logical system*, which consists of four parts:

Type theory
: A logical system has a type theory, which is a calculus of types and terms. For algebraic theories the calculus of types is trivial, since there is only one type which is not even explicitly mentioned. The terms are built from variables and basic operations.

Logic
: A logical system has a logic. A variety of different kinds of logic can be considered. Algebraic theories have a very simple kind of logic that only involves equations and equational reasoning.

Theory
: A theory is given by basic types, basic terms, and axioms. The types and the terms must be expressed in the type theory of the system, and the axioms must be expressed in the logic of the system.

Interpretations and Models
: The type theory and logic of a logical system can be interpreted in a category with sufficiently rich structure. For algebraic theories we considered categories with finite products. The interpretation is *denotational*, which means that it is defined inductively on the structure of types, terms, and logical formulas. An interpretation of a theory is a *model* if it satisfies all the axioms of the theory.

The type theory or the logic of a logical system can be very simple. When the logic is restricted to just equations between terms we usually speak of a *type theory* rather than a logical system. When the type system is trivial, so that all terms have the same type, we speak of of a *single-sorted* logic. On the other end of the spectrum are complicated logical systems in which type theory and logic are intertwined, for example in first-order logic over a dependent type theory with subset and quotient types. It can also happen that the type-theoretic and logical parts are identified, for example in Martin-Löf type theory.

Complementary to the logical system is its *categorical semantics*:

Theories are categories
: From a theory we can construct a category which expresses essentially the same information as the theory but hides syntactic details. The structure of the category reflects the underlying type theory and logic. For example, algebraic theories are categories that are sequences of finite powers of an object.

Models are functors
: A model is a functor from a theory to a category with sufficiently rich structure. The requirement that all axioms of the theory must be satisfied by a model then translates to the requirement that the model is

a functor and that it preserves the structure of the theory. For models of algebraic theories we only required that they preserve finite products, whereas functoriality ensured that all valid equations of the theory, hence also the axioms, were preserved.

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

Completeness and universal models
It is desirable for a categorical semantics to be *complete*, or to even have *universal models*. We explain what this means next.

Consider an algebraic theory $\mathbb{A}$ and an equation $u = v$ of the theory. If the equation can be proved from the axioms of the theory, then every model of the theory satisfies the equation. The converse statement is

"Every model of $\mathbb{A}$ satisfies $u = v$." $\Longrightarrow$ "$\mathbb{A}$ proves equation $u = v$." .

This property is called *semantic completeness*. Models of algebraic theories are semantically complete.

Theorem 2.1.26 (Completeness for algebraic theories) *Suppose $\mathbb{A}$ is an algebraic theory. Then there exists a category $\mathcal{A}$ and a model $U \in \mathsf{Mod}_{\mathcal{A}}(\mathbb{A})$, called the* universal model *for $\mathbb{A}$, with the property that, for every equation $u = v$ of the theory $\mathbb{A}$,*

"U satisfies $u = v$." $\iff$ *"$\mathbb{A}$ proves $u = v$."*

Therefore, categorical semantics of algebraic theories is complete.

Proof. The algebraic theory $\mathbb{A}$ is a category with finite products, so for the category $\mathcal{A}$ we can simply take $\mathbb{A}$ itself! The models of $\mathbb{A}$ in $\mathcal{A} = \mathbb{A}$ are functors $\mathbb{A} \to \mathbb{A}$ that preserve finite products. One such model is the identity functor $U = 1_{\mathbb{A}} : \mathbb{A} \to \mathbb{A}$. Clearly, the identity functor identifies two k-ary operations $f : A^k \to A$ and $g : A^k \to A$ if, and only if, $f = g$. ∎

Classically, when we speak of models of algebraic theories we have in mind models in Set. Therefore, it is a bit unsatisfactory that the universal model from the preceding theorem is not set-theoretic. Nevertheless, we can always find a universal model in a category of generalized sets, namely in a presheaf category.[4]

Proposition 2.1.27 *Let $\mathbb{A}$ be an algebraic theory. The Yoneda embedding $\mathsf{y} : \mathbb{A} \to \widehat{\mathbb{A}}$ is a universal model for $\mathbb{A}$.*

[4] Recall that the objects of a presheaf category $\mathsf{Set}^{\mathcal{C}^{\mathsf{op}}}$ are functors $\mathcal{C}^{\mathsf{op}} \to \mathsf{Set}$, which can be thought of as sets parametrized by objects of $\mathcal{C}$. In this sense they are generalized sets.

Proof. The Yoneda embedding $\mathsf{y} : \mathbb{A} \to \widehat{\mathbb{A}}$ preserves limits, and in particular finite products, hence it is a model of $\mathbb{A}$ in $\widehat{\mathbb{A}}$. Because it is a functor it satisfies all equations that are proved by $\mathbb{A}$, and because it is faithful it does not validate any equations that are not proved by $\mathbb{A}$. In other words, it is a universal model. ∎

Example 2.1.28 We consider group theory one last time. The universal group is a group that satisfies every equation that is satisfied by all groups, and no others. Let us describe it as a generalized set. Recall that the theory of a group is a category $\mathbb{G}$ whose objects are contexts $[x_1, \ldots, x_n]$, $n \in \mathbb{N}$. The carrier U of the universal group is the Yoneda embedding of the context with one variable,

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

This is a set parametrized by the objects of $\mathbb{G}$. For every $n \in \mathbb{N}$, we get a set $U_n = \mathbb{G}([x_1, \ldots, x_n], [x_1])$ that consists of all terms built from n variables, modulo equations of group theory—which is precisely the free group on n generators! Unit, inverse, and multiplication on U are defined at each stage U_n as the corresponding operations on the free group on n generators.

To summarize, the universal group is the free group on n-generators, where $n \in \mathbb{N}$ is a parameter.

Exercise 2.1.29 The universal group U is a functor $\mathbb{G}^{\mathsf{op}} \to \mathsf{Set}$. In the last example we described the object part of U. What is the action of U on morphisms?

Exercise 2.1.30 Let s be a term of group theory with variables $x_1, \ldots, x_n$. On one hand we can think of s as an element of the free group U_n, and on the other we can consider the interpretation of s in the universal group U, namely a natural transformation $Us : U^n \Longrightarrow U$. Suppose t is another term of group theory with variables $x_1, \ldots, x_n$. Show that $Us = Ut$ if, and only if, $s = t$ in the free group U_n.

2.2 Cartesian Closed Categories

2.2.1 Exponentials

We motivate the notion of exponentials with a couple of examples. Consider first the category Poset 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:

$$f \leq g \iff \forall\, x{:}P \,.\, fx \leq gx \; . \qquad (f, g : P \to Q)$$

We see that $\mathsf{Hom}(P, Q)$ is again an object of Poset, when equipped with a suitable order.

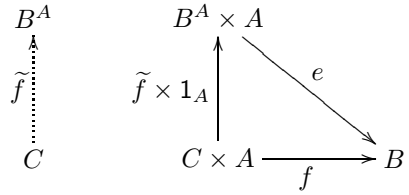
Similarly, given monoids $K, M \in \mathsf{Mon}$, there is a natural monoid structure on $\mathsf{Hom}(K, M)$ defined by

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

On the other hand, in the category of groups there seems to be no natural way of defining a group structure on the set of all homomorphisms $\mathsf{Hom}(G, H)$ between groups G and H because not all homomorphisms are bijections.

These examples suggest that there ought to be a general notion of "exponentials" in a category. Speaking informally, the idea is that for objects A and B an exponential B^A is 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 $e : B^A \times A \to B$ which evaluates a morphism $f \in B^A$ at an argument $x \in A$ to give a value $e\langle f, x\rangle \in B$.

Definition 2.2.1 In a category $\mathcal{C}$ with binary products, an *exponential* (B^A, e) of objects A and B is an object B^A together with a morphism $e : B^A \times A \to B$, 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*[5] of f, for which the following diagram commutes:



The commutativity of the above diagram means that $f = e \circ (\widetilde{f} \times 1_A)$.

The above condition is called the *universal property of exponentials*. 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 are never mentioned. The benefit of this is that the definition is completely general and applicable in categories whose objects are not sets.

In Poset the exponential Q^P of posets P and Q is the set of all monotone maps $P \to Q$, ordered pointwise. The evaluation map $e : Q^P \times P \to Q$ 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, $(\widetilde{f}z)x = f\langle z, x\rangle$. Therefore, Poset has all exponentials.

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

An object $A \in \mathcal{C}$ is *exponentiable* when the exponent B^A exists for every $B \in \mathcal{C}$. Sometimes, an object $B \in \mathcal{C}$ is called *baseable* when the exponent B^A exists for every $A \in \mathcal{C}$.

Proposition 2.2.2 *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 $\varepsilon : -^A \times A \Longrightarrow 1_{\mathcal{C}}$ 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). The object part of the right adjoint is B^A. For the morphism part, given $g : B \to C$, we define $g^A : B^A \to C^A$ to be the transpose of $g \circ \varepsilon_B$,

$$g^A = (g \circ \varepsilon_B)^\sim \ .$$

We use the formulation of adjoints by counit, cf. Proposition 1.5.5, to show that $- \times A \dashv -^A$. The counit $\varepsilon : -^A \times A \Longrightarrow 1_{\mathcal{C}}$ at B is ε_B. The naturality square for ε,

$$\begin{array}{ccc} B^A \times A & \xrightarrow{\varepsilon_B} & B \\ {\scriptstyle f^A \times 1_A}\downarrow & & \downarrow{\scriptstyle f} \\ C^A \times A & \xrightarrow[\varepsilon_C]{} & C \end{array}$$

commutes because it is just the defining property of $(f \circ \varepsilon_B)^\sim$:

$$\varepsilon_C \circ (f^A \times 1_A) = \varepsilon_C \circ ((f \circ \varepsilon_B)^\sim \times 1_A) = f \circ \varepsilon_B \ .$$

The universal property of counit ε is precisely the universal property of the exponentials. ∎

Because exponentials are expressed as right adjoints to binary products, they are determined uniquely up to isomorphism.

Example 2.2.3 Consider propositional calculus P with conjunction and implication, as in Subsection 1.5.1. Recall that P is the set of all propositions constructed from propositional variables, truth $\top$, falsehood $\bot$, conjunction $\wedge$, and implication $\Longrightarrow$. It is a preorder under the logical entailment relation $\vdash$. We saw already that implication is right adjoint to conjunction,

$$(- \times A) \dashv (A \Longrightarrow -) \ . \tag{2.2}$$

A conjunction $A \wedge B$ is a greatest lower bound of A and B, because we have, $A \wedge B \vdash A$, $A \wedge B \vdash B$, and for all propositions C,

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

Since in a preorder binary products are the same thing as greatest lower bounds, we see that conjunction is a binary product. Therefore, its right adjoint implication, is the exponential in P. The counit of the adjunction, or equivalently, the "evaluation" morphism, is the entailment

$$(A \Longrightarrow B) \wedge A \vdash B \ ,$$

which is the well known logical rule of *modus ponens*. This provides further evidence that concepts in logic arise as adjoints and their related notions.

Exercise 2.2.4 What is the unit of adjunction (2.2) in logical terms?

2.2.2 Cartesian Closed Categories

Definition 2.2.5 A *cartesian category* is a category that has finite products.

Definition 2.2.6 A *cartesian closed category (ccc)* is a category that has finite products and exponentials.

Equivalently, we could require the existence of the terminal object, binary products, and exponentials. The definition of cartesian closed categories can be phrased in terms of adjoint functors.

Proposition 2.2.7 *A category $\mathcal{C}$ is cartesian closed if, and only if, the following functors have* right *adjoints:*

$$\begin{aligned} !_{\mathcal{C}} &: \mathcal{C} \to 1 \ , \\ \Delta &: \mathcal{C} \to \mathcal{C} \times \mathcal{C} \ , \\ (- \times A) &: \mathcal{C} \to \mathcal{C} \ . \qquad (A \in \mathcal{C}) \end{aligned}$$

Proof. Here $!_{\mathcal{C}}$ is the unique functor from $\mathcal{C}$ to the terminal category 1, and Δ is the diagonal functor,

$$\Delta A = \langle A, A \rangle \ , \qquad \Delta f = f \times f \ .$$

The right adjoint of $!_{\mathcal{C}}$ is a terminal object, the right adjoint of Δ is the binary product, and the right adjoint of $- \times A$ is exponentiation by A. ∎

We give a third formulation of cartesian closed categories, in terms of equations. 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 $\pi_0^{A,B} : A \times B \to A$ and $\pi_1^{A,B} : A \times B \to B$, 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 $e_{A,B} : B^A \times A \to B$, and a morphism $\widetilde{f} : C \to B^A$ for every morphism $f : C \times A \to B$.

We usually write π_0 and π_1 instead of $\pi_0^{A,B}$ and $\pi_1^{A,B}$. For morphisms $f : A \to B$ and $g : A' \to B'$ we define $f \times g : A \times A' \to B \times B'$ to be

$$f \times g = \langle f \circ \pi_0^{A,A'}, g \circ \pi_1^{A,A'} \rangle \ .$$

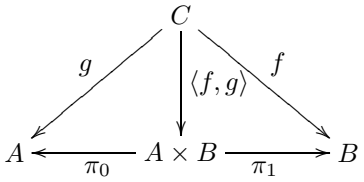
The types and morphisms satisfy the following equations:

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 $f : C \times A \to B$, $g : C \to B^A$,
$$e_{A,B} \circ (\widetilde{f} \times 1_A) = f \ , \qquad (e_{A,B} \circ (g \times 1_A))^{\sim} = g \ .$$

These equations ensure that certain diagrams commute and that morphisms which are required to exist are unique. For example, let us prove that $(A \times B, \pi_0, \pi_1)$ is the product of A and B. For $f : C \to A$ and $g : C \to B$ there exists a morphism $\langle f, g \rangle : C \to A \times B$. Equations

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

enforce the commutativity of the two triangles in the following diagram:



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.

We now look at examples of cartesian closed categories.

Example 2.2.8 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 the set of all functions from X to Y,[6]

$$Y^X = \left\{ f \subseteq X \times Y \;\middle|\; \forall\, x{:}X \,.\, \exists!\, y{:}Y \,.\, \langle x, y\rangle \in f \right\} \;,$$

and the evaluation morphism $e : Y^X \times X \to Y$ is the usual evaluation of a function at an argument, i.e., $e\langle f, x\rangle$ is the unique $y \in Y$ for which $\langle x, y\rangle \in f$.

Example 2.2.9 The category Cat of all small categories is cartesian closed. The exponential of small categories $\mathcal{C}$ and $\mathcal{D}$ is the functor category $\mathcal{D}^{\mathcal{C}}$, cf. isomorphism 1.4 on page 17.

Example 2.2.10 A presheaf category $\widehat{\mathcal{C}}$ is cartesian closed, provided $\mathcal{C}$ is small. To see what the exponential of presheaves P and Q ought to be, we use Yoneda Lemma. If Q^P exists, then by Yoneda Lemma and the adjunction $(- \times P) \dashv (-^P)$, we have for all $A \in \mathcal{C}$,

$$Q^P A \cong \mathsf{Nat}(\mathsf{y}A, Q^P) \cong \mathsf{Nat}(\mathsf{y}A \times P, Q) \;.$$

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 2.2.11 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 $\mathsf{Set}^{\cdot \rightrightarrows \cdot}$.

We have already seen that the exponential of posets P and Q is the poset Q^P of monotone functions from P to Q, ordered pointwise. Therefore Poset is cartesian closed. It is worth noting that even though the forgetful functor $U : \mathsf{Poset} \to \mathsf{Set}$ preserves finite limits, it does *not* preserve exponentials; in general $U(Q^P)$ is a proper subset of $(UQ)^{UP}$.

[6] In set theory, a function is the same thing as a functional relation.

Exercise 2.2.12 There is a full and faithful functor $I : \mathsf{Set} \to \mathsf{Poset}$. Describe it and show that it preserves finite limits as well as exponentials.

Exercise 2.2.13 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 this adjoint is *not* the binary product in Vec.

Next we consider two important categories of cartesian closed posets—frames and Heyting algebras. They play an important role in logic and frames are important for topology, too.

2.2.3 Frames

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 2.2.14 *A poset is complete if, and only if, it is cocomplete.*

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

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

■

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 $\{y_i \in P \mid i \in I\}$,

$$x \wedge \bigvee_{i\in I} y_i = \bigvee_{i\in I}(x \wedge y_i) \tag{2.3}$$

as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{x \wedge \bigvee_{i\in I} y_i \leq z}{\bigvee_{i\in I} y_i \leq (x \Rightarrow z)}}{\forall\, i{:}I\,.\,(y_i \leq (x \Rightarrow z))}}{\forall\, i{:}I\,.\,(x \wedge y_i \leq z)}}{\bigvee_{i\in I}(x \wedge y_i) \leq z}}$$

Now since $x \wedge \bigvee_{i\in I} y_i$ and $\bigvee_{i\in I}(x \wedge y_i)$ have the same upper bounds they must be equal.

Conversely, suppose the distributive law (2.3) holds. Then we can *define* $x \Rightarrow y$ to be

$$(x \Rightarrow y) = \bigvee \{z \in P \mid x \wedge z \leq y\} \ . \tag{2.4}$$

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 $x, y \in P$,

$$x \wedge (x \Rightarrow y) \leq y \ , \tag{2.5}$$

and that, for $z \in P$,

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

This implication follows directly from (2.4), and (2.5) follows from the distributive law:

$$x \wedge (x \Rightarrow y) = x \wedge \bigvee \{z \in P \mid x \wedge z \leq y\} = \bigvee \{x \wedge z \mid x \wedge z \leq y\} \leq y \ .$$

Complete cartesian closed posets are called *frames*.

Definition 2.2.15 A *frame* is a poset that is complete and cartesian closed. Equivalently, a frame is a complete poset satisfying the 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.

Example 2.2.16 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.

Exercise* 2.2.17 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,

$$N(x) = N(y) \Longrightarrow x = y . \qquad (x, y \in X)$$

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?

2.2.4 Heyting Algebras

A *lattice* is a poset that has finite limits and colimits. In other words, a lattice $(L, \leq, \wedge, \vee, 0, 1)$ is a poset $(L, \leq)$ with distinguished elements $0, 1 \in L$, and binary operations meet $\wedge$ and join $\vee$, satisfying for all $x, y, z \in L$,

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

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:

$$\begin{aligned} (x \wedge y) \wedge z &= x \wedge (y \wedge z) , & (x \vee y) \vee z &= x \vee (y \vee z) , \\ x \wedge y &= y \wedge x , & x \vee y &= y \vee x , \\ x \wedge x &= x , & x \vee x &= x , \\ 1 \wedge x &= x , & 0 \vee x &= x , \\ & x \wedge (y \vee x) = x = (x \wedge y) \vee x . \end{aligned} \qquad (2.6)$$

The partial order on L is then determined by

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

Exercise 2.2.18 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:

$$\begin{aligned} (x \vee y) \wedge z &= (x \wedge z) \vee (y \wedge z) \ , \\ (x \wedge y) \vee z &= (x \vee z) \wedge (y \vee z) \ . \end{aligned} \tag{2.7}$$

It turns out that if one distributive law holds then so does the other [Joh82, I.1.5].

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

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

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.

A Heyting algebra can be axiomatized equationally as a set H with two distinguished elements 0 and 1 and three binary operations $\wedge$, $\vee$ and $\Rightarrow$. The axioms for a Heyting algebra are the ones listed in (2.6), as well as the following ones for $\Rightarrow$:

$$\begin{aligned} (x \Rightarrow x) &= 1 \ , \\ x \wedge (x \Rightarrow y) &= x \wedge y \ , \\ y \wedge (x \Rightarrow y) &= y \ , \\ (x \Rightarrow (y \wedge z)) &= (x \Rightarrow y) \wedge (x \Rightarrow z) \ . \end{aligned} \tag{2.8}$$

For a proof, see [Joh82, I.1.10].

It turns out that every Heyting algebra is distributive [Joh82, I.1.11].

Every frame is a Heyting algebra because it has all limits and colimits, therefore also the finite ones.

2.2.5 Intuitionistic Propositional Calculus

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

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

1. Constants: $\bot, \top \in HS$.

2. Generators: if $x \in S$ then $x \in HS$.

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 it as such, to be the smallest equivalence relation satisfying axioms (2.6) and (2.8). This forces HS to be a Heyting algebra. We still need to define the action of H on morphisms: a function $f : S \to T$ is mapped to the Heyting algebra morphism $Hf : HS \to HT$ defined by

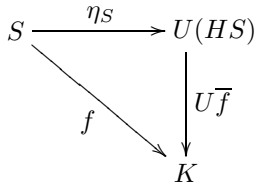
$$(Hf)\bot = \bot\,, \qquad (Hf)\bot = \bot\,, \qquad (Hf)x = fx\,,$$
$$(Hf)(\phi \star \psi) = ((Hf)\phi) \star ((Hf)\psi)\,,$$

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

There is an inclusion $\eta_S : S \to U(HS)$ of generators into the underlying set of HS. In fact we get a natural transformation $\eta : 1_{\mathsf{Set}} \Longrightarrow U \circ H$ which is 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 morphism $\overline{f} : HS \to K$ 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:



It is a unique such morphism because any two morphisms from HS which agree on generators are equal. This is proved by induction on the structure of 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 logician's point of view—normally a *calculus* consists of a language, judgments, and rules of inference—but here we want to emphasize the idea that objects of interest, even when they are syntactically constructed, are characterized by their universal properties.

Having said that, let us also describe IPC in the usual 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 $\Rightarrow$. In IPC the basic judgment 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 $u{:}A$. Logical 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 $\Gamma \vdash A$. For example, here is a proof of $A \vee B \vdash B \vee A$:

$$\frac{\dfrac{}{A \vee B \vdash A \vee B} \qquad \dfrac{\dfrac{}{A \vee B, A \vdash A}}{A \vee B, A \vdash B \vee A} \qquad \dfrac{\dfrac{}{A \vee B, B \vdash B}}{A \vee B, B \vdash B \vee A}}{A \vee B \vdash B \vee A}$$

We did not bother to label the hypotheses. 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 2.2.22

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

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

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 2.2.19 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 2.2.20 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, ...

2.2.6 Boolean Algebras

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 $x \in B$,

$$x \wedge \neg x = 0 \,, \qquad\qquad x \vee \neg x = 1 \,. \tag{2.9}$$

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

Exercise 2.2.21 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 2.2.22 Derive the following properties of negation in a *Heyting* algebra:

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

Exercise 2.2.23 The topology $\mathcal{O}X$ of a topological space X is a frame, therefore a Heyting algebra. Describe in topological language negation on $\mathcal{O}X$ and regular elements in $\mathcal{O}X$.

Exercise 2.2.24 Show that for a Heyting algebra H, the regular elements of H form a Boolean algebra $H_{\neg\neg} = \big\{x \in H \;\big|\; x = \neg\neg x\big\}$. 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*:

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

If we identify logically equivalent formulas of CPC we obtain a poset CPC ordered by logical entailment. This poset can be described by a universal property: it is the free Boolean algebra on countably many generators. The construction of a free Boolean algebra is performed just like the construction of a free Heyting algebra. The equational axioms for a Boolean algebra are the axioms for a lattice (2.6), the distributive laws (2.7), and the complement laws (2.9).

Exercise* 2.2.25 Is CPC isomorphic to the Boolean algebra $\mathsf{IPC}_{\neg\neg}$ of the regular elements of IPC?

Exercise 2.2.26 Show that in a Heyting algebra H, $\neg\neg x = x$ 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 $\forall\, x{:}H\,.\, \neg\neg x = x$ means that double negation is an order-reversing bijection $H \to H$. Therefore it transforms joins into meets and vice versa, and so *De Morgan laws* hold:

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

De Morgan laws together with $y \wedge \neg y = 0$ easily imply $y \vee \neg y = 1$. See [Joh82, I.1.11].

2.3 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. 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

$$fx = t \ .$$

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.*[7] For example, $\lambda x.\, \lambda y.\, (x + y)$ is the function which maps an argument a to the function $\lambda y.\, (a + y)$.

In an expression $\lambda x.\, t$ the variable x is *bound* in t. This means that we can rename it and we still have the same λ-abstraction. For example, $\lambda x.\, \lambda y.\, (x + y)$, $\lambda u.\, \lambda y.\, (u + y)$, and $\lambda u.\, \lambda x.\, (u + x)$ are all the same λ-abstraction. This is similar to bound variables in integrals, summations, and quantified formulas:

$$\int \frac{1}{1 + ax^2}\, dx\ , \qquad \sum_{k=1}^{\infty} \frac{1}{k^n}\ , \qquad \exists\, u{:}A\,.\, P(u, v)\ .$$

The bound variables are x, k, and u, respectively. The other variables, a, n and v, are *free*. If t is an expression then we denote by $\mathsf{FV}(t)$ the set of free variables of t. The fact that λ-abstraction binds a variable is expressed by the rules for computing $\mathsf{FV}(t)$:

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

When we rename a bound variable, or substitute for a free one, we must be careful not to *capture* any variables. For example, if we want to substitute $1 + z$ for y in the expression

$$\lambda x.\, \lambda z.\, (x + y + z)\ ,$$

then we must first rename the bound variable z, say to w, and only then substitute:

$$\lambda x.\, \lambda w.\, (x + 1 + z + w)\ .$$

If we substituted directly, we would get $\lambda x.\, \lambda z.\, (x + 1 + z + z)$, which is not what was intended because z was captured by the λ-abstraction.

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 means. 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

$$t : A\ .$$

[7] Why is the letter λ used? The notation goes back to Alonzo Church. We have it on good authority that once professor Church was sent a postcard asking him "Why λ?" He wrote the answer onto the same postcard and returned it. The answer was this: "enie menie miney mo".

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 $t \in A$. In these notes 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 *basic types*. We express the fact that A is a basic type with an axiom

$$\frac{}{A \ \texttt{type}}$$

which is read as "A is a type". There is a *unit type* `1`:

$$\frac{}{\texttt{1 type}}$$

We can also form *product types* and *function types*:

$$\frac{A \ \texttt{type} \qquad B \ \texttt{type}}{A \times B \ \texttt{type}} \qquad\qquad \frac{A \ \texttt{type} \qquad B \ \texttt{type}}{A \to B \ \texttt{type}}$$

To summarize, the set of all types is generated from the basic types and the unit type `1` by formation of product and function types. 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 the basic constants by the following grammar:

$$t ::= [\text{var}] \mid [\text{const}] \mid * \mid \langle t, t' \rangle \mid \texttt{fst}\, t \mid \texttt{snd}\, t \mid t\, t' \mid \lambda x{:}A \,.\, t$$

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 $\texttt{fst}\, t$ and $\texttt{snd}\, t$ 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 u and t are terms and x is a variable, then we obtain a new term $t[u/x]$ by *substitution* of u for x in t. This means that we replace every *free* occurrence of x in t by u. If u has any free variables, we must make sure that they are not *captured* by the bound variables in t. This is accomplished by renaming the bound variables in t so that they are disjoint from the free variables in u. Assuming that this is the case, the rules for substitution are as follows:

$$\begin{aligned}
x[u/x] &= u\\
y[u/x] &= y \quad \text{if } x \neq y\\
a[u/x] &= a \quad \text{if } a \text{ is a basic constant}\\
\langle s, t\rangle[u/x] &= \langle s[u/x], t[u/x]\rangle\\
\mathtt{fst}\,t[u/x] &= \mathtt{fst}\,(t[u/x])\\
\mathtt{snd}\,t[u/x] &= \mathtt{snd}\,(t[u/x])\\
(s\,t)[u/x] &= (s[u/x])(t[u/x])\\
(\lambda y{:}A\,.\,t)[u/x] &= \lambda y{:}A\,.\,(t[u/x]) \quad \text{if } x \neq y \text{ and } y \notin \mathsf{FV}(u)
\end{aligned}$$

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 a has a uniquely determined type A,

$$\frac{}{\Gamma \mid a : A}$$

The type of a variable is determined by the context:

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

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

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

The typing rules for pairs and projections are:

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

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,

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

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 2.3.1 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 2.3.2 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:

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

These are just the familiar axioms for a group.

Example 2.3.3 In general, any algebraic theory $\mathbb{A}$ determines a λ-theory. 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 2.3.4 The theory of a directed graph is a simply-typed theory with two basic types, `V` for vertices and `E` for edges, and two basic constant, source `src` and target `trg`,

$$\mathtt{src} : \mathtt{E} \to \mathtt{V}\,, \qquad\qquad \mathtt{trg} : \mathtt{E} \to \mathtt{V}\,.$$

There are no equations.

Example 2.3.5 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 `nat` for natural numbers, and a basic type `bool` of Boolean values,

$$\frac{}{\mathtt{nat\ type}} \qquad\qquad \frac{}{\mathtt{bool\ type}}$$

There are basic constants zero `0`, successor `succ`, the Boolean constants `true` and `false`, comparison with zero `iszero`, and for each type A the *conditional* $\mathtt{cond}_A$ and the *fixpoint* operator $\mathtt{fix}_A$. They have the following types:

$$\begin{aligned} \mathtt{0} &: \mathtt{nat} \\ \mathtt{succ} &: \mathtt{nat} \to \mathtt{nat} \\ \mathtt{true} &: \mathtt{bool} \\ \mathtt{false} &: \mathtt{bool} \\ \mathtt{iszero} &: \mathtt{nat} \to \mathtt{bool} \\ \mathtt{cond}_A &: \mathtt{bool} \to A \to A \\ \mathtt{fix}_A &: (A \to A) \to A \end{aligned}$$

The equational axioms of PCF are:

$$\begin{aligned} \cdot \mid \mathtt{iszero}\ \mathtt{0} &= \mathtt{true} : \mathtt{bool} \\ x : \mathtt{nat} \mid \mathtt{iszero}\,(\mathtt{succ}\,x) &= \mathtt{false} : \mathtt{bool} \\ u : A, t : A \mid \mathtt{cond}_A\ \mathtt{true}\ u\ t &= u : A \\ u : A, t : A \mid \mathtt{cond}_A\ \mathtt{false}\ u\ t &= t : A \\ t : A \to A \mid \mathtt{fix}_A\ t &= t\,(\mathtt{fix}_A\ t) : A \end{aligned}$$

Example 2.3.6 Another example of a λ-theory is the *theory of a reflexive type*. This theory has one basic type `D` and two constants

$$\mathtt{r} : \mathtt{D} \to \mathtt{D} \to \mathtt{D} \qquad\qquad \mathtt{s} : (\mathtt{D} \to \mathtt{D}) \to \mathtt{D}$$

satisfying the equation

$$f : \mathtt{D} \to \mathtt{D} \mid \mathtt{r}\,(\mathtt{s}\,f) = f : \mathtt{D} \to \mathtt{D} \tag{2.10}$$

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

$$x : \mathtt{D} \mid \mathtt{s}\,(\mathtt{r}\,x) = x : \mathtt{D} \tag{2.11}$$

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

2.3.1 Untyped λ-calculus

We briefly describe the *untyped λ-calculus*. It is a theory whose terms are generated by the following grammar:

$$t ::= [\text{var}] \;\big|\; t\,t' \;\big|\; \lambda x.\, t\ .$$

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:

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

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

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

The untyped λ-calculus can be translated into the theory of a reflexive type from Example 2.3.6. 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

$$\Gamma \mid u = t \tag{2.12}$$

is translated to the judgment

$$\Gamma^* \mid u^* = t^* : \mathtt{D}\,. \tag{2.13}$$

Exercise* 2.3.7 Prove that if equation (2.12) is provable then equation (2.13) is provable as well. Identify precisely at which point in your proof you need to use equations (2.10) and (2.11). Does provability of (2.13) imply provability of (2.12)?

2.4 Completeness of λ-calculus

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

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

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

The interpretation is extended to all terms in contexts 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 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
$$[\![\Gamma]\!] \xrightarrow{!_{[\![\Gamma]\!]}} 1 \xrightarrow{[\![c]\!]} [\![A]\!]$$

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, $[\![\Gamma \mid u : A]\!] = [\![\Gamma \mid t : A]\!]$. It follows that all equations provable in $\mathbb{T}$ are satisfied in the model.

In Section 2.1 we saw that algebraic theories can be viewed as categories, cf. Definition 2.1.14, and models as functors, cf. Definition 2.1.19. The same can be done with λ-theories and their models. The first step is to build a category from a λ-theory.

Given a λ-theory $\mathbb{T}$, we construct a *syntactic category* $\mathcal{S}(\mathbb{T})$ as follows. The objects of $\mathcal{S}(\mathbb{T})$ 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 terms
$$x : A \mid t : B \qquad \text{and} \qquad y : B \mid u : C$$
is the term obtained by substituting t for y in u:
$$x : A \mid u[t/y] : C \,.$$
The identity morphism on A is the term $x : A \mid x : A$.

Proposition 2.4.1 *The syntactic category $\mathcal{S}(\mathbb{T})$ built from a λ-theory is cartesian closed.*

Proof. 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 define models as functors.

Definition 2.4.2 A *model* M of a λ-theory $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$ is a functor $M : \mathcal{S}(\mathbb{T}) \to \mathcal{C}$ that preserves finite products and exponentials.

We can now proceed much as we did in the case of algebraic theories and prove that the semantics of λ-theories in cartesian closed categories is complete:

"$\mathbb{T}$ proves $\Gamma \mid t = u : A$." $\iff$ "Every model of $\mathbb{T}$ satisfies $\Gamma \mid t = u : A$."

This is proved exactly the same way as for algebraic theories. A λ-theory $\mathbb{T}$ has a *canonical interpretation* in the syntactic category $\mathcal{S}(\mathbb{T})$ 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*. It is in fact a universal model for $\mathbb{T}$ because by construction of $\mathcal{S}(\mathbb{T})$, for any terms $\Gamma \mid u : A$ and $\Gamma \mid t : A$,

"$\mathbb{T}$ proves $\Gamma \mid u = t : A$" $\iff$ "Syntactic model satisfies $\Gamma \mid u = t : A$" .

We take one step further and organize λ-theories into a category. For this we need to define a suitable notion of morphisms. 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 of consideration shows that a translation $\tau : \mathbb{T} \to \mathbb{U}$ is the same thing as a model of $\mathbb{T}$ in $\mathcal{S}(\mathbb{U})$.

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 $\mathcal{S}(\mathbb{T})$.

Definition 2.4.3 $\lambda\mathsf{Thr}$ 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
$$x : \ulcorner A \urcorner \mid \ulcorner 1_A \urcorner x = x : \ulcorner A \urcorner \, .$$
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
$$\mathtt{T} : 1 \to \ulcorner \mathtt{1} \urcorner \, ,$$
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
$$\mathbb{T} \underset{\sigma}{\overset{\tau}{\rightleftarrows}} \mathbb{U}$$

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 posses 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 $\mathcal{S}(\mathbb{T})$. This extends to a functor

$$\mathcal{S} : \lambda\mathsf{Thr} \to \mathsf{Ccc}\ ,$$

because a translation $\tau : \mathbb{T} \to \mathbb{U}$ induces a functor $\mathcal{S}(\tau) : \mathcal{S}(\mathbb{T}) \to \mathcal{S}(\mathbb{U})$ in an obvious way: a basic type $A \in \mathcal{S}(\mathbb{T})$ is mapped to the object $\tau A \in \mathcal{S}(\mathbb{U})$, 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 $\mathcal{S}(\tau)$ is defined inductively on the structure of types and terms.

Theorem 2.4.4 *The functors $\mathbb{L} : \mathsf{Ccc} \to \lambda\mathsf{Thr}$ and $\mathcal{S} : \lambda\mathsf{Thr} \to \mathsf{Ccc}$ constitute an equivalence of categories, "up to equivalence". This means that for any $\mathcal{C} \in \mathsf{Ccc}$ there is an equivalence of catgories*

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

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

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

Proof. For a small cartesian closed category $\mathcal{C}$, consider the functor $\eta_{\mathcal{C}} : \mathcal{C} \to \mathcal{S}(\mathbb{L}(\mathcal{C}))$, defined for an object $A \in \mathcal{C}$ and $f : A \to B$ in $\mathcal{C}$ by

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

To see that $\eta_{\mathcal{C}}$ is a functor, observe that $\mathbb{L}(\mathcal{C})$ proves, for all $A \in \mathcal{C}$,

$$x : \ulcorner A \urcorner \mid \ulcorner 1_A \urcorner x = x : \ulcorner A \urcorner$$

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 $X \in \mathcal{S}(\mathbb{L}(\mathcal{C}))$ 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 $\theta_{\mathcal{C}} : \mathcal{S}(\mathbb{L}(\mathcal{C})) \to \mathcal{C}$ so that $\theta_{\mathcal{C}} \circ \eta_{\mathcal{C}} \cong 1_{\mathcal{C}}$ and $\eta_{\mathcal{C}} \circ \theta_{\mathcal{C}} \cong 1_{\mathcal{S}(\mathbb{L}(\mathcal{C}))}$.

Given a λ-theory $\mathbb{T}$, we define a translation $\tau_{\mathbb{T}} : \mathbb{T} \to \mathbb{L}(\mathcal{S}(\mathbb{T}))$. 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 $\sigma_{\mathbb{T}} : \mathbb{L}(\mathcal{S}(\mathbb{T})) \to \mathbb{T}$ as follows. If $\ulcorner A \urcorner$ is a basic type in $\mathbb{L}(\mathcal{S}(\mathbb{T}))$ 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

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

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 $\mathbb{L}(\mathcal{S}(\mathbb{T}))$, 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 2.4.5 In the previous proof we defined, for each $\mathcal{C} \in \mathsf{Ccc}$, a functor $\eta_C : \mathcal{C} \to \mathcal{S}(\mathbb{L}(\mathcal{C}))$. Verify that this determines a natural transformation $\eta : 1_{\mathsf{Ccc}} \Longrightarrow \mathcal{S} \circ \mathbb{L}$. 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?

Chapter 4

Elementary Logic

Having considered type theories, we now move on to first-order logic. This is the usual logic with propositional connectives $\wedge$, $\Longrightarrow$, and $\vee$, and quantifiers $\forall$ and $\exists$. The general approach to studying logic via category theory is similar to the approach taken for type theory—we specify certain categorical structures and show how to model in them first-order logic, or a suitable fragment of it. Here adjoint functors play an imporant role, as the basic logical operations are recognized as adjoints. We also show that semantics is "functorial", which means that models of a theory are functors that preserve suitable categorical structure. Finally, we construct classifying categories, which are the logical counterparts of syntactic models of type theories.

Let us demonstrate our approach with an example. In Section 2.1 we studied algebraic theories, in which we can express properties in terms of equations, for example commutativity

$$x \cdot y = y \cdot x \ .$$

As we saw, such equations could be interpreted in any category with finite products. This provided a large scope for categorical semantics of algebraic theories. However, there are many relevant properties of algebraic structures which cannot with equations. Consider the statement that a group $(G, e, \cdot, ^{-1})$ has no non-trivial roots of unit,

$$\forall\, x{:}G\,.\,(x \cdot x = e \Longrightarrow x = e) \ . \tag{4.1}$$

This is a first-order logic statement which cannot be rewritten as a system of equations. To see what its categorical interpretation ought to be, we look at its usual set-theoretic interpretation. Each of subformulas, $x \cdot x = e$ and $x = e$, determines a subset of G:

$$\begin{array}{ccc} \{x \in G \mid x \cdot x = e\} & \rightarrowtail & \{x \in G \mid x = e\} \\ & {\scriptstyle i} \searrow \quad \swarrow {\scriptstyle j} & \\ & G & \end{array}$$

The implication $x \cdot x = e \Longrightarrow x = e$ holds when $\{x \in G \mid x \cdot x = e\}$ is contained in $\{x \in G \mid x = e\}$. In categorical language we say that the mono i factors through the mono j. Observe also that such a factorization is necessarily a mono and is unique, if it exists. The defining formulas of the subsets $\{x \in G \mid x \cdot x = e\}$ and $\{x \in G \mid x = e\}$ are equations and so the subsets can be constructed as equalizers:

$$\{x \in G \mid x \cdot x = e\} \rightarrowtail G \underset{e \circ !_G}{\overset{\langle 1_G, 1_G\rangle}{\rightrightarrows}} G \times G \xrightarrow{\cdot} G$$

$$\{x \in G \mid x = e\} \rightarrowtail G \underset{e \circ !_G}{\overset{1_G}{\rightrightarrows}} G$$

The second equalizer is in fact isomorphic to the morphism $e : 1 \to G$. Thus we can interpret condition (4.1) in any category with products and equalizers, in other words a category with finite limits.[1] This allows us to define the notion of a group without roots of unit in any category $\mathcal{C}$ with finite limits as an object G with morphisms $e : 1 \to G$, $m : G \times G \to G$ and $i : G \to G$ such that (G, e, m, i) is a group in $\mathcal{C}$, and the equalizer of $m \circ \langle 1_G, 1_G\rangle$ and $e \circ !_G$ factors through $e : 1 \to G$.

The aim of this chapter is to analyze how such examples can be studied in general. We want to relate first-order logic and fragments of it to categorical structures that are suitable for interpreation of the logic.

4.1 First-order Theories

A *first-order theory* $\mathbb{L}$ consists of an underlying *type theory* and a *fragment of first-order logic*. Recall from Chapter 2 that a simple type theory is given by a set of basic types, a set of basic constants together with their types, rules for forming types, rules and axioms for deriving typing judgments

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid t : B \ ,$$

expressing that term t has type B in typing context $x_1{:}A_1$, …, $x_n{:}A_n$, and a set of axioms and rules of inference which tell us which equations between terms

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid t = u : B \ .$$

are valid.

A fragment of first-order logic is given by a set of *basic relation symbols* together with a specification of which part of first-order logic is being considered.

[1] We are *not* claiming that finite limits suffice for an interpretation of arbitrary formulas built from universal quantifiers and implications. The formula at hand has a very special form $\forall x \,.\, (\varphi(x) \Rightarrow \psi(x))$, where $\varphi(x)$ and $\psi(x)$ do not contain further $\forall$ or $\Rightarrow$.

Each basic relation symbol has a specified *signature* $(A_1, \ldots, A_n)$, which specifies the types of its arguments. The *arity* of a relation symbol is the number of arguments it takes. The judgment

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid \phi \ \mathtt{pred}$$

states that ϕ is a well-formed formula in typing context $x_1{:}A_1, \ldots, x_n{:}A_n$. For each basic relation symbol R with signature $(A_1, \ldots, A_n)$ there is an inference rule

$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid R(t_1, \ldots, t_n) \ \mathtt{pred}}$$

Depending on what fragment of first-order logic is included, there might be other rules for forming logical formulas. For example, if equality is present, then for each type A there is a rule

$$\frac{\Gamma \mid t : A \qquad \Gamma \mid u : A}{\Gamma \mid t =_A u \ \mathtt{pred}}$$

and if conjunction is present, then there is a rule

$$\frac{\Gamma \mid \varphi \ \mathtt{pred} \qquad \Gamma \mid \psi \ \mathtt{pred}}{\Gamma \mid \varphi \wedge \psi \ \mathtt{pred}}$$

Other such rules will be given when we come to the study of particular logical operations.

The basic logical judgment of a first-order theory is *logical entailment*

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid \varphi_1, \ldots, \varphi_m \vdash \psi$$

which states that in the typing context $x_1{:}A_1, \ldots, x_n{:}A_n$ the hypotheses φ_1, $\ldots$, φ_m entail ψ. It is understood that the terms appearing in the formulas are well-typed in the typing context, and that formulas $\varphi_1, \ldots, \varphi_m, \psi$ are part of the fragment of the logic of $\mathbb{T}$. In the theory $\mathbb{T}$ there are axioms and inference rules for deriving valid judgments. When the fragment contains equality, we replace the type-theoretic equality judgments

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid t = u : B$$

with the logical statements

$$x_1{:}A_1, \ldots, x_n{:}A_n \mid \cdot \vdash t =_B u \ .$$

The subscript at the equality sign indicates the type at which the equality is taken.

Fragments of first-order logic can be given in various ways. The usual one is to select a subset of logical connectives and quantifiers. There are several standard fragments:

1. *Full first-order logic* is built from logical operations

$$= \quad \top \quad \bot \quad \wedge \quad \vee \quad \Longrightarrow \quad \forall \quad \exists \,.$$

   Equality is more properly considered as a family of binary relation symbols, one for each type of the underlying type calculus. Thus each type A is equipped with its own equality $=_A$.

2. *Lex*[2] *logic* is the fragment built from

$$= \quad \top \quad \wedge \,.$$

3. *Regular logic* is the fragment built from

$$= \quad \top \quad \wedge \quad \exists \,.$$

4. *Coherent logic* is the fragment built from

$$= \quad \top \quad \wedge \quad \exists \quad \bot \quad \vee \,.$$

5. A *geometric formula* is a formula of the form

$$\forall\, x{:}A \,.\, (\varphi \Rightarrow \psi) \ ,$$

   where φ and ψ are coherent formulas.

The names for these fragments come from the names of categorical structures in which they are interpreted.

The well formed terms and the formulas of a first-order theory $\mathbb{T}$ constitute its *language*. It may seem that we are doing things backwards, because we should have spoken of first-order languages before we spoke of first-order theories. While this may be possible for simple theories, it certainly becomes hard to do when we consider complicated theories in which types and logical formulas are intertwined. In such cases the typing judgments and logical entailments might be given by a mutual recursive definition. In order to find out whether a given term is well formed, we might have to prove a logical statement. In everyday mathematics this occurs all the time, for example, to show that the term $\int_0^\infty f$ denotes a real number, it may be necessary to prove that $f : \mathbb{R} \to \mathbb{R}$ is an integrable function and that the integral has a finite value. This is why it does not always make sense to strictly differentiate a language from a theory.[3]

In order to emphasize the logical part of first-order theories, we are going to limit attention to only two very simple kinds of type theory. A *single-sorted* first-order theory has as its underlying type theory a single type A and for each $k \in \mathbb{N}$ a set of basic k-ary function symbols. The rules for typing judgments are:

[2] The name "lex" comes from "left exact", which refers to a category with finite limits.

[3] However, it *does* make sense to distinguish syntax from theory. Rules of substitution and the behaviour of free and bound variables are syntactic considerations, for example.

1. Variables in contexts:

$$\frac{}{x_1{:}A, \ldots, x_n{:}A \mid x_i : A}$$

2. For each basic function symbol f of arity k, there is an inference rule

$$\frac{\Gamma \mid t_1 : A \qquad \cdots \qquad \Gamma \mid t_n : A}{\Gamma \mid f\langle t_1, \ldots, t_n\rangle : A}$$

This is essentially like an algebraic theory, except that the type is explicitly named. In addition a single-sorted first-order theory may contain relation symbols, formulas, axioms, and rules of inference which an algebraic theory does not.

A slight generalization of a single-sorted theory is a *multi-sorted* one. Its underlying type theory is given by a set of types, and a set of basic function symbols. Each function symbol f has a *signature* $(A_1, \ldots, A_n; B)$, where n is the arity of f and $A_1, \ldots, A_n, B$ are types. The rules for typing judgments are:

1. Variables in contexts:

$$\frac{}{x_1{:}A_1, \ldots, x_n{:}A_n \mid x_i : A_i}$$

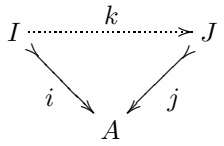
2. For each basic function symbol f with signature $(A_1, \ldots, A_n; B)$, there is an inference rule

$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid f\langle t_1, \ldots, t_n\rangle : B}$$

We often write suggestively $f : A_1 \times \cdots \times A_n \to B$ to indicate that $(A_1, \ldots, A_n; B)$ is the signature of f. However, this does not mean that $A_1 \times \cdots \times A_n \to B$ is a type! A multi-sorted type theory does *not* have any type forming operations, such as $\times$ and $\to$.[4]

4.2 The Subobject Functor

Let A be an object in a category $\mathcal{C}$. If $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ are monos into A, we say that i is smaller than j, and write $i \leq j$, when there exists a morphism $k : I \to J$ such that the following diagram commutes:



[4]However, a simple type theory can be viewed as a multi-sorted type theory, if we forget that it has type forming operations.

If such a k exists then it is unique, and it is always mono. The class $\mathsf{Mono}(A)$ of all monos into A is preordered by $\leq$. Let $\mathsf{Sub}(A)$ be its poset reflection. Thus the elements of $\mathsf{Sub}(A)$ are equivalence classes of monos, where monos $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ are equivalent when $i \leq j$ and $j \leq i$. The induced relation $\leq$ on $\mathsf{Sub}(A)$ is a partial order.

We have to be a bit careful with the formation of $\mathsf{Sub}(A)$, since it is defined as a quotient of a *class* $\mathsf{Mono}(A)$. In many particular cases the general construction by quotients can be avoided. If we is demonstrate that the preorder $\mathsf{Mono}(A)$ is equivalent, as a category, to a poset P then we can take $\mathsf{Sub}(A) = P$. At any rate, we usually require that $\mathsf{Sub}(A)$ is small.

Definition 4.2.1 A category is *essentially small* when it is equivalent to a small category.

Definition 4.2.2 A category $\mathcal{C}$ is *well-powered* when, for all $A \in \mathcal{C}$, the category $\mathsf{Mono}(A)$ is equivalent to a small poset. In other words, for every $A \in \mathcal{C}$, $\mathsf{Sub}(A)$ is a small category.

We often speak of subobjects as if they were monos rather than equivalence classes of monos. It is understood that we mean the subobjects represented by monos and not the monos themselves. Sometimes we refer to a mono $i : I \rightarrowtail A$ by its domain I only, even though the object I itself does not determine the morphism i. Hopefully this will not cause confusion, as it is always going to be clear which mono is meant to go along with the object I.

The assignment $A \mapsto \mathsf{Sub}(A)$ is the object part of the *subobject functor*

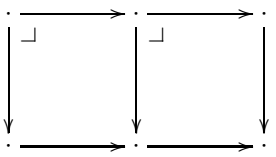
$$\mathsf{Sub} : \mathcal{C}^{\mathsf{op}} \to \mathsf{Poset} .$$

The morphism part of Sub is pullback. More precisely, given a morphism $f : A \to B$, let $\mathsf{Sub}(f) = f^* : \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ be the monotone map which maps the subobject $[i : I \rightarrowtail B]$ to the subobject $[f^*i : f^*I \rightarrowtail A]$, where $f^*i : f^*I \rightarrowtail A$ is a pullback of i along f:

$$\begin{array}{ccc} f^*I & \longrightarrow & I \\ {\scriptstyle f^*i}\downarrow & \lrcorner & \downarrow{\scriptstyle i} \\ A & \xrightarrow[f]{} & B \end{array}$$

Recall that a pullback of a mono is again mono, so this definition makes sense. We need to verify that $\mathsf{Sub}(1_A) = 1_{\mathsf{Sub}(A)}$ and $\mathsf{Sub}(g \circ f) = \mathsf{Sub}(f) \circ \mathsf{Sub}(g)$. The first equations is obvious and the second one follows from the following lemma.

Lemma 4.2.3 *Suppose both squares in the following diagram are pullbacks:*



Then the outer rectangle is a pullback diagram as well.

Proof. This is left as an exercise in diagram chasing. ∎

Pullbacks are only determined up to isomorphism, but this does not cause any problems because isomorphic monos represent the same subobject.

In categorical semantics, a formula

$$x : A \mid \varphi \texttt{ pred}$$

is interpreted as a subobject

$$[\![x : A \mid \varphi]\!] \rightarrowtail [\![A]\!]$$

Logical operations then correspond to operations on the poset $\mathsf{Sub}(A)$. Therefore, the structure of $\mathsf{Sub}(A)$ determines which logical connectives can be interpreted. If $\mathsf{Sub}(A)$ is a Heyting algebra, then we can interpret intuitionistic propositional calculus, cf. Subsection 2.2.4, but if $\mathsf{Sub}(A)$ only has binary meets then all that can be interpreted is $\top$ and $\wedge$. We will work this out in detail in the following sections.

Another use of subobjects appears in the interpretation of substitution. There are two kinds of substitution, in a term and in a formula. We may substitute a term $x : A \mid t : B$ for a variable y in a term $y : B \mid u : C$ to obtain a new term $x : A \mid u[t/y] : C$. If t and u are interpreted as morphisms

$$[\![A]\!] \xrightarrow{[\![t]\!]} [\![B]\!] \xrightarrow{[\![u]\!]} [\![C]\!]$$

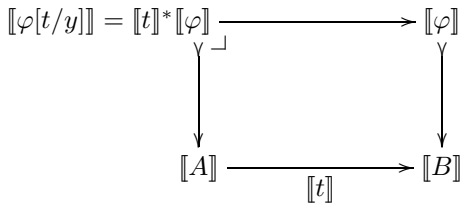
then $u[t/y]$ is interpreted as their composition:

$$[\![x : A \mid u[t/y] : C]\!] = [\![y : B \mid u : C]\!] \circ [\![x : A \mid t : B]\!] \,.$$

Thus, *substitution in a term is composition.*

The second kind of substitution occurs when we substitute a term $x : A \mid t : B$ for a variable y in a formula $y : B \mid \varphi$ `pred` to obtain a new formula $x : A \mid \varphi[t/y]$ `pred`. If t is interpreted as a morphism $[\![t]\!] : [\![A]\!] \to [\![B]\!]$ and φ is interpreted as a subobject $[\![\varphi]\!] \rightarrowtail [\![B]\!]$ then the interpretation of $\varphi[t/y]$ is the

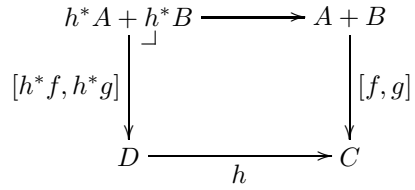
pullback of $[\![\varphi]\!]$ along $[\![t]\!]$:



Thus, *substitution in a formula is pullback,*

$$[\![x : A \mid \varphi[t/y] \ \texttt{pred}]\!] = \mathsf{Sub}([\![x : A \mid t : B]\!])[\![y : B \mid \varphi \ \texttt{pred}]\!] \,.$$

Because substitution is a very basic operation, we should expect that categorical structure needed to interpret various logical operations must behave well with respect to pullbacks. We say that a categorical property or structure is *stable (under pullbacks)* if it is preserved by pullbacks. For example, a category $\mathcal{C}$ has stable binary coproducts if it has binary coproducts and that the pullback of $[f, g] : A + B \to C$ along $h : D \to C$ is (isomorphic to) $[h^*f, h^*g] : h^*A + h^*B \to D$:



4.3 Lex Logic

As a first example we look at the categorical semantics of lex logic over a multi-sorted type theory with unit type $\mathtt{1}$. We have already dealt with multi-sorted type theories and the axioms for the unit type, so we concentrate on the logic instead.

The logical connectives are $=$, $\top$, and $\wedge$. The rules for forming formulas are as follows:

1. Substitution:
$$\frac{\Gamma \mid t : A \qquad \Gamma, x : A \mid \varphi \ \texttt{pred}}{\Gamma \mid \varphi[t/x] \ \texttt{pred}}$$

2. Weakening:
$$\frac{\Gamma \mid \varphi \ \texttt{pred}}{\Gamma, x : A \mid \varphi \ \texttt{pred}}$$

3. For each basic relation symbol R with signature $(A_1, \ldots, A_n)$ there is a rule
$$\frac{\Gamma \mid t_1 : A_1 \qquad \cdots \qquad \Gamma \mid t_n : A_n}{\Gamma \mid R(t_1, \ldots, t_n) \ \texttt{pred}}$$

4. The logical constant $\top$ is a formula:

$$\frac{}{\Gamma \mid \top \ \texttt{pred}}$$

5. For each type A, there is a rule

$$\frac{\Gamma \mid t : A \qquad \Gamma \mid u : A}{\Gamma \mid t =_A u \ \texttt{pred}}$$

6. Conjunction:

$$\frac{\Gamma \mid \varphi \ \texttt{pred} \qquad \Gamma \mid \psi \ \texttt{pred}}{\Gamma \mid \varphi \wedge \psi \ \texttt{pred}}$$

The rules of inference are:

1. Weakening:

$$\frac{\Gamma \mid \Psi \vdash \varphi}{\Gamma \mid \Psi, \psi \vdash \varphi}$$

2. Substitution:

$$\frac{\Gamma \mid t : A \qquad \Gamma, x{:}A \mid \Psi \vdash \varphi}{\Gamma \mid \Psi[t/x] \vdash \varphi[t/x]}$$

3. Cut:

$$\frac{\Gamma \mid \Psi \vdash \theta \qquad \Gamma \mid \Psi, \theta \vdash \varphi}{\Gamma \mid \Psi \vdash \varphi}$$

4. Axioms:

$$\frac{}{\Gamma \mid \psi_1, \ldots, \psi_n \vdash \psi_i} \qquad (1 \leq i \leq n)$$

5. Truth:

$$\frac{}{\Gamma \mid \Psi \vdash \top}$$

6. Equality:

$$\frac{}{\Gamma \mid \Psi \vdash t =_A t} \qquad \frac{\Gamma \mid \Psi \vdash t =_A u \qquad \Gamma \mid \Psi \vdash \varphi[t/z]}{\Gamma \mid \Psi \vdash \varphi[u/z]}$$

7. Conjunction:

$$\frac{\Gamma \mid \Psi \vdash \varphi \qquad \Gamma \mid \Psi \vdash \psi}{\Gamma \mid \Psi \vdash \varphi \wedge \psi} \qquad \frac{\Gamma \mid \Psi \vdash \varphi \wedge \psi}{\Gamma \mid \Psi \vdash \psi} \qquad \frac{\Gamma \mid \Psi \vdash \varphi \wedge \psi}{\Gamma \mid \Psi \vdash \varphi}$$

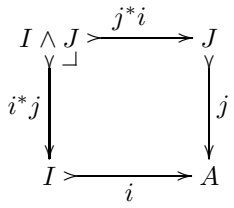
Exercise 4.3.1 Derive symmetry and transitivity of equality:

$$\frac{\Gamma \mid \Psi \vdash t =_A u}{\Gamma \mid \Psi \vdash u =_A t} \qquad \frac{\Gamma \mid \Psi \vdash t =_A u \qquad \Gamma \mid \Psi \vdash u =_A v}{\Gamma \mid \Psi \vdash t =_A v}$$

Before we embark on semantics of lex logic, we note a couple of useful propositions.

Proposition 4.3.2 *If a category $\mathcal{C}$ has pullbacks then, for every $A \in \mathcal{C}$, $\mathsf{Sub}(A)$ has finite limits.*

Proof. The poset $\mathsf{Sub}(A)$ has finite limits if it has a top object and binary meets. The top object of $\mathsf{Sub}(A)$ is the subobject $[1_A : A \to A]$. The meet of subobjects $i : I \rightarrowtail A$ and $j : J \rightarrowtail A$ is the subobject $i \wedge j = i \circ (i^* j) = j \circ (j^* i) : I \wedge J \rightarrowtail A$ obtained by pullback, as in the following diagram:



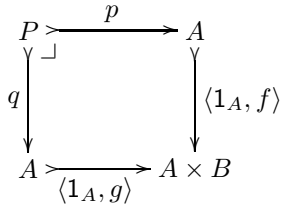
It is easy to verify that $I \wedge J$ is the infimum of I and J. ∎

Proposition 4.3.3 *If a category has finite products and pullbacks of monos along monos then it has all finite limits.*

Proof. It is sufficient to show that the category has equalizers. To construct the equalizer of parallel arrows $f : A \to B$ and $g : A \to B$, first observe that the arrows

$$A \xrightarrow{\langle 1_A, f \rangle} A \times B \qquad\qquad A \xrightarrow{\langle 1_A, g \rangle} A \times B$$

are monos because the projection $\pi_0 : A \times B \to A$ is their left inverse. Therefore, we may construct the pullback



The morphisms p and q coincide because $\langle 1_A, f \rangle$ and $\langle 1_A, g \rangle$ have a common left inverse π_0:

$$p = 1_A \circ p = \pi_0 \circ \langle 1_A, f \rangle \circ p = \pi_0 \circ \langle 1_A, f \rangle \circ q = 1_A \circ q = q \ .$$

Let us show that $p : P \to A$ is the equalizer of f and g. First, p equalizes f and g,

$$f \circ p = \pi_1 \circ \langle 1_A, f \rangle \circ p = \pi_1 \circ \langle 1_A, g \rangle \circ q = g \circ q = g \circ p \ .$$

If $k : K \to A$ also equalizes f and g then

$$\langle 1_A, f \rangle \circ k = \langle k, f \circ k \rangle = \langle k, g \circ k \rangle = \langle 1_A, g \rangle \circ k \ ,$$

therefore by the universal property of the constructed pullback there exists a unique factorization $\overline{k} : K \to P$ such that $k = p \circ \overline{k}$, as required. ∎

We now explain how lex logic is interpreted in a finitely complete category $\mathcal{C}$. Let $\mathbb{T}$ be a multi-sorted lex theory. Recall that the type theory of $\mathbb{T}$ is specified by a set of sorts (types) and a set of basic function symbols together with their signatures. The logic is given by a set of basic relation symbols with their signatures, and a set of axioms in the form of logical entailments

$$\Gamma \mid \Psi \vdash \varphi \,.$$

An interpretation of $\mathbb{T}$ in $\mathcal{C}$ is given by the following data, where Γ stands for a typing context $x_1{:}A_1, \ldots, x_n{:}A_n$, and Ψ stands for a sequence of formulas $\psi_1, \ldots, \psi_k$:

1. A sort A is interpreted as an object $[\![A]\!]$.

2. The unit sort $\mathtt{1}$ is interpreted as the terminal object 1.

3. A typing context $x_1{:}A_1, \ldots, x_n{:}A_n$ is interpreted as the product $[\![A_1]\!] \times \cdots \times [\![A_n]\!]$. The empty context is interpreted as the terminal object 1.

4. A basic function symbol f with signature $(A_1, \ldots, A_m; B)$ is interpreted as a morphism $[\![f]\!] : [\![A_1]\!] \times \cdots [\![A_m]\!] \to [\![B]\!]$.

5. A term in a context $\Gamma \mid t : B$ is interpreted as a morphism $[\![\Gamma \mid t : B]\!] : [\![\Gamma]\!] \to [\![B]\!]$, as follows:

   (a) A variable $x_0{:}A_1, \ldots, x_n{:}A_n \mid x_i : A_i$ is interpreted as the i-th projection $\pi_i : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![A_i]\!]$.

   (b) The interpretation of $\Gamma \mid * : \mathtt{1}$ is the unique morphism $!_{[\![\Gamma]\!]} : [\![\Gamma]\!] \to 1$.

   (c) A composite term $\Gamma \mid f(t_1, \ldots, t_m) : B$, where f is a basic function symbol with signature $(A_1, \ldots, A_m; B)$, is interpreted as the composition

   $$[\![\Gamma]\!] \xrightarrow{\langle [\![t_1]\!], \ldots, [\![t_m]\!] \rangle} [\![A_1]\!] \times \cdots \times [\![A_m]\!] \xrightarrow{[\![f]\!]} [\![B]\!]$$

   Here $[\![t_i]\!]$ is shorthand for $[\![\Gamma \mid t_i : A_i]\!]$.

6. A basic relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted as a subobject $[\![R]\!] \in \mathsf{Sub}([\![A_1]\!] \times \cdots \times [\![A_n]\!])$.

7. A formula in a context $\Gamma \mid \varphi$ `pred` is interpreted as a subobject $[\![\Gamma \mid \varphi]\!] \in \mathsf{Sub}([\![\Gamma]\!])$. The details are given below.

8. A logical entailment $\Gamma \mid \Psi \vdash \varphi$ is interpreted as an inequality $[\![\Psi]\!] \leq [\![\varphi]\!]$ in $\mathsf{Sub}([\![\Gamma]\!])$. Here the interpretation of Ψ is the infimum $[\![\Psi]\!] = [\![\psi_1]\!] \wedge \cdots \wedge [\![\psi_k]\!]$. The empty sequence of hypotheses is interpreted as the maximal subobject $1_{[\![\Gamma]\!]} : [\![\Gamma]\!] \rightarrowtail [\![\Gamma]\!]$.

It remains to explain how formulas are interpreted as subobjects. As was explained in the previous section, a formula formed by substitution is interpreted as the left-hand side of the pullback:

$$\begin{array}{ccc} [\![\Gamma \mid \varphi[t/x]]\!] & \longrightarrow & [\![\Gamma, x{:}A \mid \varphi]\!] \\ \downarrow & & \downarrow \\ [\![\Gamma]\!] & \xrightarrow[\langle 1_\Gamma, [\![t]\!]\rangle]{} & [\![\Gamma]\!] \times [\![A]\!] \end{array}$$

A formula formed by weakening is interpreted as pullback along a projection:

$$\begin{array}{ccc} [\![\Gamma, x{:}A \mid \varphi]\!] & \longrightarrow & [\![\Gamma \mid \varphi]\!] \\ \downarrow & & \downarrow i \\ [\![\Gamma]\!] \times [\![A]\!] & \xrightarrow[\pi_0]{} & [\![\Gamma]\!] \end{array}$$

This pullback can be computed and the interpretation of $[\![\Gamma, x{:}A \mid \varphi]\!]$ turns out to be the subobject

$$[\![\Gamma \mid \varphi]\!] \times [\![A]\!] \rightarrowtail [\![\Gamma]\!] \times [\![A]\!]$$

An atomic formula $\Gamma \mid R(t_1, \ldots, t_m)$, where R is a basic relation symbol with signature $(A_1, \ldots, A_m)$ is interpreted as the left-hand side of the pullback pullback

$$\begin{array}{ccc} [\![R(t_1, \ldots, t_m)]\!] & \longrightarrow & [\![R]\!] \\ \downarrow & & \downarrow \\ [\![\Gamma]\!] & \xrightarrow[\langle [\![t_1]\!], \ldots, [\![t_m]\!]\rangle]{} & [\![A_1]\!] \times \cdots \times [\![A_m]\!] \end{array}$$

The logical constant $\top$ is interpreted as the maximal subobject:

$$[\![\Gamma \mid \top]\!] = [1_{[\![\Gamma]\!]} : [\![\Gamma]\!] \to [\![\Gamma]\!]] \,.$$

An equation $\Gamma \mid t =_A u$ `pred` is interpreted as the subobject represented by the equalizer of $[\![\Gamma \mid t : A]\!]$ and $[\![\Gamma \mid u : A]\!]$:

$$[\![\Gamma \mid t =_A u]\!] \rightarrowtail [\![\Gamma]\!] \underset{[\![u]\!]}{\overset{[\![t]\!]}{\rightrightarrows}} [\![A]\!]$$

By Proposition 4.3.2, each $\mathsf{Sub}(A)$ is a poset with binary meets. Thus we interpret a conjunction $\Gamma \mid \varphi \wedge \psi$ `pred` as the infimum of subobjects

$$[\![\Gamma \mid \varphi \wedge \psi]\!] = [\![\Gamma \mid \varphi]\!] \wedge [\![\Gamma \mid \psi]\!] \,.$$

This concludes the description of an interpretation of lex theory $\mathbb{T}$ in a lex category $\mathcal{C}$.

When we deal with many interpretations at once we name them M, N, ..., and subscript the semantic brackets accordingly, $[\![\Gamma]\!]_M$, $[\![\Gamma]\!]_N$, ...

If $\Gamma \mid \Psi \vdash \psi$ is a logical entailment in $\mathbb{T}$ such that $[\![\Gamma \mid \Psi]\!]_M \leq [\![\Gamma \mid \varphi]\!]_M$ holds in an interpretation M, then we say that M *satisfies* or *models* $\Gamma \mid \Psi \vdash \psi$ and write

$$M \models \Gamma \mid \Psi \vdash \psi \,.$$

An interpretation M is a *model* of $\mathbb{T}$ if it satisfies all the axioms of $\mathbb{T}$.

Theorem 4.3.4 (Soundness of lex logic) *If a lex theory $\mathbb{T}$ proves an entailment*

$$\Gamma \mid \Psi \vdash \psi$$

then every model M of $\mathbb{T}$ satisfies the entailment:

$$M \models \Gamma \mid \Psi \vdash \psi \,.$$

Proof. The proof proceeds by induction on the proof of the entailment. In the following we usually omit the typing context Γ to simplify notation, and all inequalities are interpreted in $\mathsf{Sub}([\![\Gamma]\!])$. We consider all possible last steps in the proof of the entailment:

1. Weakening: if $[\![\Psi]\!] \leq [\![\varphi]\!]$ then

$$[\![\Psi, \psi]\!] = [\![\Psi]\!] \wedge [\![\psi]\!] \leq [\![\Psi]\!] \leq [\![\varphi]\!] \,.$$

2. Substitution: recall that substitution is interpreted by pullback so that $[\![\varphi[t/x]]\!] = \langle 1_{[\![\Psi]\!]}, [\![t]\!]\rangle^* [\![\varphi]\!]$ and $[\![\Psi[t/x]]\!] = \langle 1_{[\![\Psi]\!]}, [\![t]\!]\rangle^* [\![\Psi]\!]$. Because

$$\langle 1_{[\![\Psi]\!]}, [\![t]\!]\rangle^* : \mathsf{Sub}([\![\Psi]\!]) \to \mathsf{Sub}([\![\Psi]\!] \times [\![A]\!])$$

is a functor it is a monotone map, therefore $[\![\Psi]\!] \leq [\![\varphi]\!]$ implies

$$\langle 1_{[\![\Psi]\!]}, [\![t]\!]\rangle^* [\![\Psi]\!] \leq \langle 1_{[\![\Psi]\!]}, [\![t]\!]\rangle^* [\![\varphi]\!] \,.$$

3. Cut: if $[\![\Psi]\!] \leq [\![\theta]\!]$ and $[\![\Psi, \theta]\!] \leq [\![\varphi]\!]$ then

$$[\![\Psi]\!] = [\![\Psi]\!] \wedge [\![\theta]\!] = [\![\Psi, \theta]\!] \leq [\![\varphi]\!] \,.$$

4. Axioms: trivially

$$[\![\psi_1, \ldots, \psi_k]\!] = [\![\psi_1]\!] \wedge \cdots \wedge [\![\psi_k]\!] \leq [\![\psi_i]\!] \,.$$

5. Truth: trivially $[\![\Psi]\!] \leq [\![\top]\!]$.

6. Equality: an axiom $t =_A t$ is satisfied because an equalizer of $[\![t]\!]$ with itself is the maximal subobject:

$$[\![\Psi]\!] \leq [1_{[\![\Gamma]\!]} : [\![\Gamma]\!] \to [\![\Gamma]\!]] = [\![t =_A t]\!] .$$

For the other axiom, suppose $[\![\Psi]\!] \leq [\![t =_A u]\!]$ and $[\![\Psi]\!] \leq [\![\varphi[t/z]]\!]$. It suffices to show $[\![t =_A u]\!] \wedge [\![\varphi[t/z]]\!] \leq [\![\varphi[u/z]]\!]$ for then

$$[\![\Psi]\!] \leq [\![t =_A u]\!] \wedge [\![\varphi[t/z]]\!] \leq [\![\varphi[u/z]]\!] .$$

The interpretation of $P = [\![t =_A u]\!] \wedge [\![\varphi[t/z]]\!]$ is obtained by two successive pullbacks, as in the following diagram:

$$\begin{array}{ccccc} P & \longrightarrow & [\![\varphi[t/z]]\!] & \longrightarrow & [\![\varphi]\!] \\ \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow \\ [\![t =_A u]\!] & \xrightarrow[e]{} & [\![\Gamma]\!] & \xrightarrow[\langle 1_\Gamma, [\![t]\!]\rangle]{} & [\![\Gamma]\!] \times [\![A]\!] \end{array}$$

Here e is the equalizer of $[\![u]\!]$ and $[\![t]\!]$. Observe that e equalizes $\langle 1_{[\![\Gamma]\!]}, [\![t]\!]\rangle$ and $\langle 1_{[\![\Gamma]\!]}, [\![u]\!]\rangle$ as well:

$$\langle 1_{[\![\Gamma]\!]}, [\![t]\!]\rangle \circ e = \langle e, [\![t]\!] \circ e\rangle = \langle e, [\![u]\!] \circ e\rangle = \langle 1_{[\![\Gamma]\!]}, [\![u]\!]\rangle \circ e .$$

Therefore, if we replace $\langle 1_{[\![\Gamma]\!]}, [\![t]\!]\rangle$ with $\langle 1_{[\![\Gamma]\!]}, [\![u]\!]\rangle$ in the above diagram, the outer rectangle still commutes. By the universal property of the pullback

$$\begin{array}{ccc} [\![\Gamma \mid \varphi[u/z]]\!] & \longrightarrow & [\![\Gamma, z{:}A \mid \varphi]\!] \\ \downarrow & \lrcorner & \downarrow \\ [\![\Gamma]\!] & \xrightarrow[\langle 1_\Gamma, [\![u]\!]\rangle]{} & [\![\Gamma]\!] \times [\![A]\!] \end{array}$$

it follows that P factors through $[\![\varphi[u/z]]\!]$, as required.

7. The rules for conjunction clearly hold because by the definition of infimum $[\![\Psi]\!] \leq [\![\varphi \wedge \psi]\!]$ if, and only if, $[\![\Psi]\!] \leq [\![\varphi]\!]$ and $[\![\Psi]\!] \leq [\![\psi]\!]$.

■

Example 4.3.5 The theory of a poset is a lex theory. There is one basic sort $\mathtt{P}$ and one binary relation symbol $\leq$ with signature $(\mathtt{P}, \mathtt{P})$. The axioms are the familiar axioms for reflexivity, transitivity, and antisymmetry:

$$\begin{aligned} x{:}\mathtt{P} \mid \cdot &\vdash x \leq x \\ x{:}\mathtt{P}, y{:}\mathtt{P}, z{:}\mathtt{P} \mid x \leq y,\ y \leq z &\vdash x \leq z \\ x{:}\mathtt{P}, y{:}\mathtt{P} \mid x \leq y,\ y \leq x &\vdash x =_{\mathtt{P}} y \end{aligned}$$

A poset in a lex category $\mathcal{C}$ is given by an object P, which is the interpretation of the sort $\mathtt{P}$, and a subobject $r : R \rightarrowtail P \times P$, which the interpretation of $\leq$, such that the axioms are satisfied. As an example we spell out when the reflexivity axiom is satisfied. The interpretation of $x{:}P \mid x \leq x$ is obtained by the following pullback:

$$\begin{array}{ccc} [\![x \leq x]\!] & \rightarrowtail & R \\ \downarrow & \lrcorner & \downarrow{\scriptstyle r} \\ P & \xrightarrow[\delta_P]{} & P \times P \end{array}$$

where $\delta_P = \langle 1_P, 1_P \rangle$ is the diagonal. The first axiom is satisfied when $[\![x \leq x]\!] = P$, which happens if, and only if, δ_P factors through r. Therefore, reflexivity can be expressed as follows: there exists a "reflexivity" morphism $\rho : P \to R$ such that $r \circ \rho = \delta_P$. Equivalently, morphisms $\pi_0 \circ r$ and $\pi_1 \circ r$ have a common right inverse ρ.

4.3.1 Subset types

Let us consider whether the theory of a category is a lex theory. We begin by expressing the definition of a category so that it can be interpreted in any lex category $\mathcal{C}$. An *internal category* in $\mathcal{C}$ consists of an *object of morphisms* C_1, an *object of objects* C_0, and *domain*, *codomain*, and *identity* morphisms,

$$\mathsf{dom} : C_1 \to C_0 \,, \qquad \mathsf{cod} : C_1 \to C_0 \,, \qquad \mathsf{id} : C_0 \to C_1 \,.$$

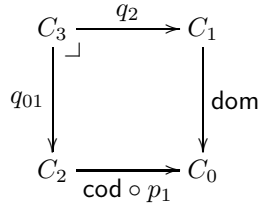
There is also a *composition* morphism $c : C_2 \to C_1$, where C_2 is obtained by the pullback

$$\begin{array}{ccc} C_2 & \xrightarrow{p_1} & C_1 \\ \downarrow{\scriptstyle p_0} & \lrcorner & \downarrow{\scriptstyle \mathsf{dom}} \\ C_1 & \xrightarrow[\mathsf{cod}]{} & C_0 \end{array}$$

The following equations must hold:

$$\begin{aligned} \mathsf{dom} \circ i = 1_{C_0} &= \mathsf{cod} \circ i \,, \\ \mathsf{cod} \circ p_1 = \mathsf{cod} \circ c \,, &\qquad \mathsf{dom} \circ p_0 = \mathsf{dom} \circ c \,. \\ c \circ \langle 1_{C_1}, i \circ \mathsf{dom} \rangle = 1_{C_1} &= c \circ \langle i \circ \mathsf{cod}, 1_{C_1} \rangle \,, \end{aligned}$$

The first two equations state that the domain and codomain of an identity morphism 1_A are both A. The second equation states that $\mathsf{cod}\,(f \circ g) = \mathsf{cod}\, f$ and the third one that $\mathsf{dom}\,(f \circ g) = \mathsf{dom}\, g$. The fourth equation states that $f \circ 1_{\mathsf{dom}\, f} = f = 1_{\mathsf{cod}\, f} \circ f$. It remains to express associativity of composition. For this purpose we construct the pullback



The object C_3 can be thought of as the set of triples of morphisms (f, g, h) such that $\mathsf{cod}\, f = \mathsf{dom}\, g$ and $\mathsf{cod}\, g = \mathsf{dom}\, h$. We denote $q_0 = p_0 \circ q_{01}$ and $q_1 = p_1 \circ q_{01}$. The morphisms $q_0, q_1, q_2 : C_3 \to C_1$ are like three projections which select the first, second, and third element of a triple, respectively. With this notation we can write $q_{01} = \langle q_0, q_1 \rangle_{C_2}$ because q_{01} is the unique morphism such that $p_0 \circ q_{01} = q_0$ and $p_1 \circ q_{01} = q_1$. The subscript C_2 reminds us that the "pair" $\langle q_0, q_1 \rangle_{C_2}$ is obtained by the universal property of the pullback C_2.

Morphisms $c \circ q_{01} : C_3 \to C_1$ and $q_2 : C_3 \to C_1$ factor through the pullback C_2 because

$$\mathsf{cod} \circ c \circ q_{01} = \mathsf{cod} \circ p_1 \circ q_0 = \mathsf{dom} \circ q_2 \ .$$

Thus let $r : C_3 \to C_2$ be the unique factorization for which $p_0 \circ r = c \circ q_{01}$ and $p_1 \circ r = q_2$. Because p_0 and p_1 are like projections from C_2 to C_1, morphism r can be thought of as a pair of morphisms, so we write $r = \langle c \circ q_{01}, q_2 \rangle_{C_2}$. Morphism $c \circ \langle c \circ q_{01}, q_2 \rangle_{C_2} : C_3 \to C_1$ corresponds to the operations $\langle f, g, h \rangle \mapsto (f, g) \circ h$, whereas the morphism corresponding to $\langle f, g, h \rangle \mapsto f \circ (g \circ h)$ is obtained in a similar way and is equal to

$$c \circ \langle q_0, c \circ \langle q_1, q_2 \rangle_{C_2} \rangle_{C_2} : C_3 \to C_1 \ .$$

Thus associativity is expressed by the equation

$$c \circ \langle c \circ \langle q_0, q_1 \rangle_{C_2}, q_2 \rangle_{C_2} = c \circ \langle q_0, c \circ \langle q_1, q_2 \rangle_{C_2} \rangle_{C_2} \ .$$

Example 4.3.6 An internal category in Set is a small category.

We have successfully formulated the theory of a category so that it makes sense in any lex category. In fact, the definition of an internal category refers only to certain pullbacks, hence the notion of an internal category makes sense in any category with pullbacks. However, if we try to formulate it as a multi-sorted lex theory, there is a problem. Obviously, there ought to be a basic sort of objects $\mathtt{C}_0$ and a basic sort of morphisms $\mathtt{C}_1$. There are also basic function symbols with signatures

$$\mathtt{dom} : (\mathtt{C}_1; \mathtt{C}_0) \qquad \mathtt{cod} : (\mathtt{C}_1; \mathtt{C}_0) \qquad \mathtt{id} : (\mathtt{C}_0, \mathtt{C}_1) \ .$$

However, it is not clear what the signature for composition should be. It is not $(\mathtt{C}_1, \mathtt{C}_1; \mathtt{C}_1)$ because composition is undefined for non-composable pairs of morphisms. We might be tempted to postulate another basic sort $\mathtt{C}_2$ but then we would have no way of stating that $\mathtt{C}_2$ is the pullback of `dom` and `cod`. And even if we somehow axiomatized the fact that $\mathtt{C}_2$ is a pullback, we would then still have to formalize the object $\mathtt{C}_3$ of composable triples, $\mathtt{C}_4$ of composable quadruples, and so on. What we lack is the ability to define the type $\mathtt{C}_2$ as a *subset type* of $\mathtt{C}_1 \times \mathtt{C}_1$.

In order to remedy the situation we need to use a richer type theory, namely one that allows *simple subset types*. We explain what these are. The formation rule for simple subset types is

$$\frac{x{:}A \mid \varphi \ \mathtt{pred}}{\{x{:}A \mid \varphi\} \ \mathtt{type}}$$

We can think of $\{x{:}A \mid \varphi\}$ as the subset of all those $x : A$ that satisfy φ. Note that we did not allow an arbitrary context Γ to be present. This means that we cannot define subset types that depend on parameters, which why they are called "simple".

Inference rules for subset types are as follows:

$$\frac{\Gamma \mid t : \{x{:}A \mid \varphi\}}{\Gamma \mid \mathtt{in}_\varphi\, t : A} \qquad \frac{\Gamma \mid t : \{x{:}A \mid \varphi\}}{\Gamma \mid \cdot \vdash \varphi[t/x]} \qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \cdot \vdash \varphi[t/x]}{\Gamma \mid \mathtt{rs}_\varphi\, t : \{x{:}A \mid \varphi\}}$$

$$\frac{\Gamma, x{:}A \mid \Psi, \varphi \vdash \theta}{\Gamma, y{:}\{x{:}A \mid \varphi\} \mid \Psi[\mathtt{in}_\varphi\, y/x] \vdash \theta[\mathtt{in}_\varphi\, y/x]}$$

The first rule states that a term t of subset type $\{x{:}A \mid \varphi\}$ can be converted to a term $\mathtt{in}_\varphi\, t$ of type A. We can think of the constant $\mathtt{in}_\varphi$ as the *inclusion* $\mathtt{in}_\varphi : \{x{:}A \mid \varphi\} \to A$. The second rule states that every term of a subset type $\{x : A \mid \varphi\}$ satisfies the defining predicate φ. The third rule states that a term t of type A which satisfies φ can be converted to a term $\mathtt{rs}_\varphi\, t$ of type $\{x{:}A \mid \varphi\}$. A good way to think of the constant $\mathtt{rs}_\varphi$ is as a partially defined *restriction*, or a type-casting operations, $\mathtt{rs}_\varphi : A \rightharpoonup \{x{:}A \mid \varphi\}$.[5] The last rule tells us how to replace a variable x of type A and an assumption φ about it with a variable y of type $\{x{:}A \mid \varphi\}$ and remove the assumption. Note that this is a two-way rule.

There are two more axioms that relate inclusions and restrictions:

$$\frac{\Gamma \mid t : \{x{:}A \mid \varphi\}}{\Gamma \mid \cdot \vdash \mathtt{rs}_\varphi\, (\mathtt{in}_\varphi\, t) = t} \qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \cdot \vdash \varphi[t/x]}{\Gamma \mid \cdot \vdash \mathtt{in}_\varphi\, (\mathtt{rs}_\varphi\, t) = t\ .}$$

In an informal discussion it is customary for the inclusions and restrictions to be omitted, or at least for the subscript φ to be missing.[6]

[5] Inclusions and restrictions are like type-casting operations in some programming languages. For example in Java, an inclusion corresponds to an (implicit) type cast from a class to its superclass, whereas a restriction corresponds to a type cast from a class to a subclass. Must I write that Java is a registered trademark of Sun Microsystems?

[6] Strictly speaking, even the notation $\mathtt{in}_\varphi\, t$ is imprecise because it does not indiciate that ϕ

Exercise 4.3.7 Suppose $x{:}A \mid \psi$ and $x{:}A \mid \varphi$ are formulas. Show that

$$x{:}A \mid \psi \vdash \varphi$$

is provable if, and only if, $\{x{:}A \mid \psi\}$ factors through $\{x{:}A \mid \varphi\}$, which means that there exists a term k,

$$y : \{x{:}A \mid \psi\} \mid k : \{x{:}A \mid \varphi\} ,$$

such that

$$y : \{x{:}A \mid \psi\} \mid \cdot \vdash \mathtt{in}_\psi\, y =_A \mathtt{in}_\varphi\, k$$

is provable. Show also that k is determined uniquely up to provable equality.

Example 4.3.8 We are now able to formulate the theory of a category as a lex theory whose underlying type theory has product types and subset types. The basic types are the type of objects $\mathtt{C}_0$ and the type of morphisms $\mathtt{C}_1$. We define the type $\mathtt{C}_2$ to be

$$\mathtt{C}_2 \equiv \{p : C_1 \times C_1 \mid \mathtt{cod}(\mathtt{fst}\, p) = \mathtt{dom}(\mathtt{snd}\, p)\} .$$

The basic function symbols and their signatures are:

$$\mathtt{dom} : \mathtt{C}_1 \to \mathtt{C}_0 , \qquad \mathtt{cod} : \mathtt{C}_1 \to \mathtt{C}_0 , \qquad \mathtt{id} : \mathtt{C}_0 \to \mathtt{C}_1 , \qquad \mathtt{c} : \mathtt{C}_2 \to \mathtt{C}_1 .$$

The axioms are:

$$\begin{aligned}
a : \mathtt{C}_0 \mid \cdot &\vdash \mathtt{dom}(\mathtt{id}(a)) = a \\
a : \mathtt{C}_0 \mid \cdot &\vdash \mathtt{cod}(\mathtt{id}(a)) = a \\
f : \mathtt{C}_1, g : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g) &\vdash \mathtt{dom}(\mathtt{c}(\mathtt{rs}\, \langle f, g\rangle)) = f \\
f : \mathtt{C}_1, g : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g) &\vdash \mathtt{cod}(\mathtt{c}(\mathtt{rs}\, \langle f, g\rangle)) = g \\
f : \mathtt{C}_1 \mid \cdot &\vdash \mathtt{c}(\mathtt{rs}\, \langle \mathtt{id}(\mathtt{dom}(f)), f\rangle) = f \\
f : \mathtt{C}_1 \mid \cdot &\vdash \mathtt{c}(\mathtt{rs}\, \langle f, \mathtt{id}(\mathtt{cod}(f))\rangle) = f
\end{aligned}$$

Lastly, the associativity axiom is

$$\begin{aligned}
f : \mathtt{C}_1, g : \mathtt{C}_1, h : \mathtt{C}_1 \mid \mathtt{cod}(f) = \mathtt{dom}(g), \mathtt{cod}(g) = \mathtt{dom}(h) &\vdash \\
\mathtt{c}(\mathtt{rs}\, \langle \mathtt{c}(\mathtt{rs}\, \langle f, g\rangle), h\rangle) &= \mathtt{c}(\mathtt{rs}\, \langle f, \mathtt{c}(\mathtt{rs}\, \langle g, h\rangle)\rangle) .
\end{aligned}$$

This notation is quite unreadable. If we write $g \circ f$ instead of $\mathtt{c}(\mathtt{rs}\, \langle f, g\rangle)$ then the axioms take on a more familiar form. For example, associativity is just $h \circ (g \circ f) = (h \circ g) \circ f$. However, we need to remember that we may form the term $g \circ f$ only if we first prove $\mathtt{dom}(g) = \mathtt{cod}(f)$.

---

stands in the context $x : A$. The correct notation would be $\mathtt{in}_{(x:A|\varphi)}\, t$, where x is bound in the subscript. A similar remark holds for $\mathtt{rs}_\varphi\, t$.

A subset type $\{x{:}A \mid \varphi\}$ is interpreted as the domain of a monomorphism representing $x{:}A \mid \varphi$:

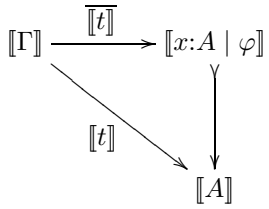
$$[\![\{x{:}A \mid \varphi\}]\!] \rightarrowtail [\![A]\!] \quad \text{via } [\![x{:}A \mid \varphi]\!]$$

Some care must be taken here because monos representing a given subobject are only determined up to isomorphism. We assume that a suitable canonical choice of monos can be made.

An inclusion $\Gamma \mid \mathtt{in}_\varphi\, t : A$ is interpreted as the composition

$$[\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![\{x{:}A \mid \varphi\}]\!] \rightarrowtail [\![A]\!] \quad \text{via } [\![x{:}A \mid \varphi]\!]$$

A restriction $\Gamma \mid \mathtt{rs}_\varphi\, t : \{x{:}A \mid \varphi\}$ is interpreted as the unique $\overline{[\![t]\!]}$ which makes the following diagram commute:



Exercise 4.3.9 Formulate and prove a soundness theorem for subset types. Pay attention to the interpretation of restrictions, where you need to show unique existence of $\overline{[\![t]\!]}$.

4.3.2 Completeness of Lex Logic

4.4 Quantifiers

The categorical semantics of quantification is one of the central features of the subject, and quite possibly one of the nicest contributions of categorical logic to the field of logic. You might expect that the quantifiers $\forall$ and $\exists$ are "just a big conjunction and disjunction", respectively. In fact the Polish school of algebraic logicians worked to realize this point of view—but categorical logic shows how quantifiers are treated algebraically as adjoint functors to give a much more satisfactory theory.

Let us first recall the rules of inference for quantifiers. The formation rules are:

$$\frac{\Gamma, x{:}A \mid \varphi \ \mathtt{pred}}{\Gamma \mid (\forall\, x{:}A\,.\,\varphi) \ \mathtt{pred}} \qquad \frac{\Gamma, x{:}A \mid \varphi \ \mathtt{pred}}{\Gamma \mid (\exists\, x{:}A\,.\,\varphi) \ \mathtt{pred}}$$

The variable x is bound in $\forall\, x{:}A\,.\,\varphi$ and $\exists\, x{:}A\,.\,\varphi$. If x and y are distinct variables and x does not occur freely in the term t then substitution of t for y commutes

with quantification over x:

$$(\exists\, x{:}A \,.\, \varphi)[t/y] = \exists\, x{:}A \,.\, (\varphi[t/y]) \ ,$$
$$(\forall\, x{:}A \,.\, \varphi)[t/y] = \forall\, x{:}A \,.\, (\varphi[t/y]) \ .$$

For each quantifier we have a two-way rule of inference:

$$\frac{\Gamma, x{:}A \mid \Psi \vdash \varphi}{\Gamma \mid \Psi \vdash \forall\, x{:}A \,.\, \varphi} \qquad \frac{\Gamma, x{:}A \mid \Psi, \varphi \vdash \theta}{\Gamma \mid \Psi, (\exists\, x{:}A \,.\, \varphi) \vdash \theta}$$

Note that the last rule implicitly imposes the usual condition that x must not occur freely in Ψ or θ because Ψ and θ are supposed to be well formed in context Γ, which does not contain x.

Exercise 4.4.1 A common way of stating the inference rules for quantifiers is as follows. For the universal quantifier, the introduction and elimination rules are

$$\frac{\Gamma, x{:}A \mid \Psi \vdash \varphi}{\Gamma \mid \Psi \vdash \forall\, x{:}A \,.\, \varphi} \qquad \frac{\Gamma \mid t : A \qquad \Gamma \mid \Psi \vdash \forall\, x{:}A \,.\, \varphi}{\Gamma \mid \Psi \vdash \varphi[t/x]}$$

The introduction rule for existential quantifier is

$$\frac{\Gamma \mid t : A \qquad \Gamma \mid \Psi \vdash \varphi[t/x]}{\Gamma \mid \Psi \vdash \exists\, x{:}A \,.\, \varphi}$$

and the elimination rule is

$$\frac{\Gamma \mid \Psi \vdash \exists\, x{:}A \,.\, \varphi \qquad \Gamma, x{:}A \mid \Psi, \varphi \vdash \theta}{\Gamma \mid \Psi \vdash \theta}$$

Note that these rules implicitly impose a requirement that x does not occur in Γ and that it does not occur freely in Ψ because the context $\Gamma, x{:}A$ must be well formed and the hypotheses Ψ must be well formed in context Γ. Show that these rules can be derived from the ones above, and vice versa. Of course, you may also use the inference rules for lex logic, cf. page 99.

In order to discover what the semantics of existential quantifier ought to be, we look at the following instance of the two-way rule for quantifiers:

$$\frac{y{:}B, x{:}A \mid \varphi \vdash \theta}{y{:}B \mid \exists\, x{:}A \,.\, \varphi \vdash \theta} \tag{4.2}$$

First observe that this rule implicitly requires

$$y{:}B, x{:}A \mid \varphi \ \texttt{pred} \qquad y{:}B \mid \theta \ \texttt{pred} \qquad y{:}B \mid (\exists\, x{:}A \,.\, \varphi) \ \texttt{pred}$$

Therefore the interpretations of φ, θ, and $\exists\, x{:}A\,.\,\varphi$ are subobjects

$$\begin{aligned}[\![y{:}B, x{:}A \mid \varphi]\!] &\in \mathsf{Sub}([\![B]\!] \times [\![A]\!])\ ,\\ [\![y{:}B \mid \theta]\!] &\in \mathsf{Sub}([\![B]\!])\ ,\\ [\![y{:}B \mid \exists\, x{:}A\,.\,\varphi]\!] &\in \mathsf{Sub}([\![B]\!])\ .\end{aligned}$$

In fact, θ appears twice, once in the context $y{:}B$ and once in the context $y{:}B, x{:}A$. The later instance is obtained from the former by a weakening rule

$$\frac{y{:}B \mid \theta \ \texttt{pred}}{y{:}B, x{:}A \mid \theta \ \texttt{pred}}$$

The interpretation of weakening is pullback along a projection, cf. page 102, as in the following pullback diagram:

$$\begin{array}{ccc} [\![y{:}B, x{:}A \mid \theta]\!] & \longrightarrow & [\![y{:}B \mid \theta]\!] \\ \downarrow & \lrcorner & \downarrow \\ [\![B]\!] \times [\![A]\!] & \xrightarrow[\pi_0]{} & [\![B]\!] \end{array}$$

Thus we have

$$[\![y{:}B, x{:}A \mid \theta]\!] = \pi_0^*[\![y{:}B \mid \theta]\!] = [\![y{:}B \mid \theta]\!] \times [\![A]\!]\ .$$

We want to interpret existential quantification $\exists x{:}A$ as a suitable functor $\exists_A : \mathsf{Sub}([\![B]\!] \times [\![A]\!]) \to \mathsf{Sub}([\![B]\!])$ so that

$$[\![y{:}B \mid \exists\, x{:}A\,.\,\varphi]\!] = \exists_A[\![y{:}B, x{:}A \mid \varphi]\!]\ .$$

The two-way rule (4.2) is then interpreted as a two-way inequality rule

$$\frac{[\![y{:}B, x{:}A \mid \varphi]\!] \le \pi_0^*[\![y{:}B \mid \theta]\!]}{\exists_A[\![y{:}B, x{:}A \mid \varphi]\!] \le [\![y{:}B \mid \theta]\!]}$$

If we replace the interpretations of φ and θ by general subobjects $S \in \mathsf{Sub}([\![B]\!] \times [\![A]\!])$ and $T \in \mathsf{Sub}([\![B]\!])$, we obtain

$$\frac{S \le \pi_0^* T}{\exists_A S \le T}$$

This is nothing but an adjunction between $\exists_A$ and π_0^*! Therefore, *existential quantification is left-adjoint to weakening*:

$$\exists_A \dashv \pi_0^*\ .$$

A dual argument shows that *universal quantification is right-adjoint to weakening*:

$$\pi_0^* \dashv \forall_A\ .$$

Let us see how all this works for the usual interpretation in Set. A predicate $y{:}B, x{:}A \mid \varphi$ corresponds to a subset $\varphi \subseteq B \times A$, and $y{:}B \mid \theta$ corresponds to a subset $\theta \subseteq B$. Weakening of θ is the subset $\pi_0^*\theta = \theta \times A \subseteq B \times A$. Then we have

$$\exists_A\varphi = \left\{ y \in B \;\middle|\; \exists\, x{:}A\,.\, \langle x, y\rangle \in \varphi \right\} \subseteq B\,,$$
$$\forall_A\varphi = \left\{ y \in B \;\middle|\; \forall\, x{:}A\,.\, \langle x, y\rangle \in \varphi \right\} \subseteq B\,.$$

A moment's thought convinces us that with this interpretation we really have

$$\frac{\varphi \subseteq \theta \times A}{\exists_A\varphi \subseteq \theta} \qquad\qquad \frac{\theta \times A \subseteq \varphi}{\theta \subseteq \forall_A\varphi}$$

The unit of the adjunction $\exists_A \dashv \pi_0^*$ amounts to the inequality

$$\varphi \subseteq (\exists_A\varphi) \times A\,, \tag{4.3}$$

and the universal property of the unit says that $\exists_A\varphi$ is the smallest set satisfying (4.3). Similarly, the counit of the adjunction $\pi_0^* \dashv \forall_A$ is just the inequality

$$(\forall_A\varphi) \times A \subseteq \varphi\,, \tag{4.4}$$

and the universal property of the counit says that $\forall_A\varphi$ is the largest set satisfying (4.4). Figure 4.1 shows the geometric meaning of existential and universal quantification.

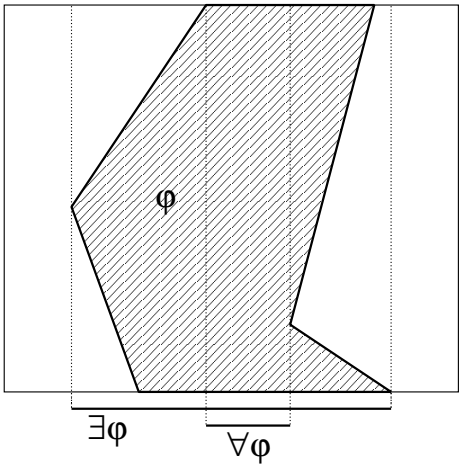


Figure 4.1: $\exists\varphi$ and $\forall\varphi$

Exercise 4.4.2 What do the universal properties of the counit of $\exists_A \dashv \pi_0^*$ and the unit of $\pi_0^* \dashv \forall_A$ say?

The weakening functor π_0^* is a special case of a pullback functor $f^* : \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ for a morphism $f : B \to A$. This gives us the idea that we may regard the left and the right adjoint to f^* as a kind of generalized existential and universal quantifier.

We may also be tempted to *define* quantifiers as left and right adjoints to pullback functors. However there is a bit more to quantifiers than that—we are still missing the important *Beck-Chevalley condition*.

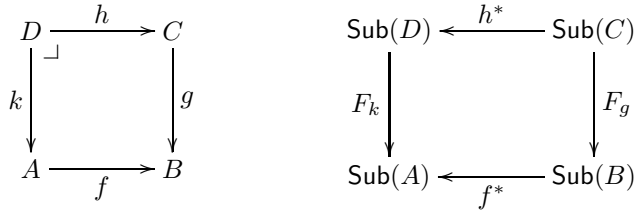
4.4.1 Beck-Chevalley Condition

Recall that quantification commutes with substitution, as long as no variables are captured by the quantifier. Thus if $\Gamma \mid t : B$ and $\Gamma, y{:}B, x{:}A \mid \varphi$ `pred` then

$$(\exists\, x{:}A\,.\,\varphi)[t/y] = \exists\, x{:}A\,.\,(\varphi[t/y]) \ .$$
$$(\forall\, x{:}A\,.\,\varphi)[t/y] = \forall\, x{:}A\,.\,(\varphi[t/y]) \ .$$

If semantics of quantifiers is to be sound, the interpretation of these equations must be valid. Because substitution of a term in a formula is interpreted as pullback this means that quantifiers must be *stable* under pullbacks. This is known as the *Beck-Chevalley condition*.

Definition 4.4.3 A family of functors $F_f : \mathsf{Sub}(A) \to \mathsf{Sub}(B)$ parametrized by morphisms $f : A \to B$ is said to satisfy the *Beck-Chevalley condition* when for every pullback on the left-hand side, the right-hand square commutes:



Definition 4.4.4 A lex category $\mathcal{C}$ has *existential quantifiers* if, for every $f : A \to B$, the left adjoint $\exists_f \dashv f^*$ exists and it satisfies the Beck-Chevalley condition. Similarly, $\mathcal{C}$ has *universal quantifiers* if the right adjoints $f^* \dashv \forall_f$ exist and they satisfy the Beck-Chevalley condition.

To convince ourselves that Beck-Chevalley condition is what we want, we spell it out in the case of a substitution into an existentially quantified formula. In order to keep the notation simple we omit the semantic brackets $[\![-]\!]$. Suppose

we have a term $\Gamma \mid t : B$ and a formula $\Gamma, y{:}B, x{:}A \mid \varphi$ `pred`. The diagram

$$\begin{array}{ccc} \Gamma \times A & \xrightarrow{\langle \pi_0, t \circ \pi_0, \pi_1 \rangle} & \Gamma \times B \times A \\ {\scriptstyle \pi_0^{\Gamma,A}}\downarrow & \lrcorner & \downarrow{\scriptstyle \pi_0^{\Gamma,B,A}} \\ \Gamma & \xrightarrow[\langle 1_\Gamma, t \rangle]{} & \Gamma \times B \end{array}$$

is a pullback. By Beck-Chevalley condition for $\exists$, the following square commutes:

$$\begin{array}{ccc} \mathsf{Sub}(\Gamma \times A) & \xleftarrow{\langle \pi_0, t \circ \pi_0, \pi_1 \rangle^*} & \mathsf{Sub}(\Gamma \times B \times A) \\ {\scriptstyle \exists_A^{\Gamma,A}}\downarrow & & \downarrow{\scriptstyle \exists_A^{\Gamma,B,A}} \\ \mathsf{Sub}(\Gamma) & \xleftarrow[\langle 1_\Gamma, t \rangle^*]{} & \mathsf{Sub}(\Gamma \times B) \end{array}$$

Therefore, for $\Gamma, y{:}B, x{:}A \mid \varphi$ `pred`,

$$\begin{aligned} [\![(\exists\, x{:}A\,.\,\varphi)[t/y]]\!] = \langle 1_\Gamma, t \rangle^* (\exists_A^{\Gamma,B,A} [\![\varphi]\!]) = \\ \exists_A^{\Gamma,A} (\langle \pi_0, t \circ \pi_0, \pi_1 \rangle^* [\![\varphi]\!]) = [\![\exists\, x{:}A\,.\,(\varphi[t/y])]\!] \,. \end{aligned}$$

This is precisely the equation we wanted.

Exercise 4.4.5 In Set we can identify $\mathsf{Sub}(-)$ with powersets because $\mathsf{Sub}(X) \cong \mathcal{P}X$. Then quantifiers along a function $f : A \to B$ are functions

$$\exists_f : \mathcal{P}A \to \mathcal{P}B \,, \qquad\qquad \forall_f : \mathcal{P}A \to \mathcal{P}B \,.$$

Verify that

$$\begin{aligned} \exists_f U = f_* U = \big\{ b \in B \;\big|\; \exists\, a{:}A\,.\,(fa = b \wedge a \in U) \big\} \,, \\ \forall_f U = \big\{ b \in B \;\big|\; \forall\, a{:}A\,.\,(fa = b \Longrightarrow a \in U) \big\} \,. \end{aligned}$$

Thus $\exists_f U$ is just the usual direct image of U by f. But have you seen $\forall_f U$ before? It can also be written as $\forall_f U = \big\{ b \in B \;\big|\; f^* \{b\} \subseteq U \big\}$. What is the meaning of $\exists_q$ and $\forall_q$ when $q : A \to A/\!\sim$ is a canonical quotient map that maps an element $x \in A$ to its equivalence class $qx = [x]$ under an equivalence relation $\sim$ on A?

4.4.2 Universal Quantifiers in LCCC's

Recall that a cartesian closed category is a category that has products and exponentials. We consider those categories for which every slice is cartesian closed.

Definition 4.4.6 A category $\mathcal{C}$ is *locally cartesian closed (lccc)* when it has a terminal object and every slice $\mathcal{C}/A$ is cartesian closed.

A slice category $\mathcal{C}/A$ always has a terminal object, namely the identity morphism $1_A : A \to A$.

Proposition 4.4.7 *A category $\mathcal{C}$ has pullbacks if, and only if, every slice $\mathcal{C}/A$ has binary products.*

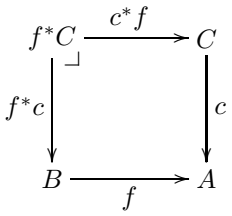
Proof. This is obvious, since the universal property of pullbacks in $\mathcal{C}$ over an object A is exactly the universal property of products in $\mathcal{C}/A$. ∎

Thus a locally cartesian closed category has all finite limits because it has a terminal object and pullbacks. In addition, a locally cartesian closed category is cartesian closed because $\mathcal{C} \cong \mathcal{C}/1$.

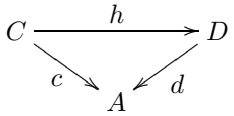
We describe how exponentials in a slice $\mathcal{C}/A$ can be computed in terms of *change of base functors* and *dependent products*. Given a morphism $f : B \to A$ in $\mathcal{C}$, the "change of base along f" is a construction

$$f^* : \mathcal{C}/A \to \mathcal{C}/B \,,$$

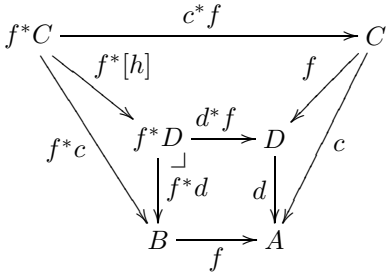
which maps $c : C \to A$ to $f^*c : f^*C \to B$, as in the pullback



and it maps a morphism



to the unique $f^*[h] : f^*C \to f^*D$ which makes the following diagram commutative:[7]



[7] We use the funny notation $f^*[h]$ to distinguish the action of f^* on morphisms in $\mathcal{C}/A$ from the usual pullback of an object in $\mathcal{C}/A$ along f.

The mapping $f^* : \mathcal{C}/A \to \mathcal{C}/B$ is usually referred to as the "change of base functor f^*", even though in general it is only a *pseudofunctor*. If f^* were a functor, it would have to satisfy the equation

$$f^*[k \circ h] = f^*[k] \circ f^*[h]$$

but in general it only satisfies the *isomorphism*

$$f^*[k \circ h] \cong f^*[k] \circ f^*[h] \ .$$

Thus f^* is a "functor up to isomorphism". For certain purposes this is a serious technical nuisance, most notably in dependent type theory. However, for our aims f^* will do just fine because we will only be concerned with adjunctions and other "up to isomorphism" constructions. For example, we can safely speak of left and right adjoints to the "functor" f^* because adjunctions are natural isomorphisms between functors, not equations. For presicse definition of pseudofunctors and related notions see [Bor94a, Definition 7.5.1].

Exercise 4.4.8 Show that the change of base functor f^* always has a left adjoint $\Sigma_f : \mathcal{C}/B \to \mathcal{C}/A$, called a *dependent sum along* f. It maps an object $c : C \to B$ to the object $\Sigma_f c = f \circ c : C \to A$, and a morphism $h : C \to D$ with domain $c : C \to B$ and codomain $d : D \to B$ in $\mathcal{C}/B$ to the morphism $h : C \to D$ with domain $f \circ c : C \to A$ and codomain $f \circ d : D \to A$ in $\mathcal{C}/A$.

A right adjoint to f^*, when it exists, is called a *dependent product along* f,

$$\Pi_f : \mathcal{C}/B \to \mathcal{C}/A \ .$$

Now an exponential of $b : B \to A$ and $c : C \to A$ in $\mathcal{C}/A$ can be computed in terms of Π_b and b^*. For any $d : D \to A$, we have $b \times_A d = (b^* d) \circ b = \Sigma_b(b^* d)$, hence

$$\cfrac{\cfrac{\cfrac{b \times_A d \to c}{\Sigma_b(b^* d) \to c}}{b^* d \to b^* c}}{d \to \Pi_b(b^* c)}$$

Therefore, $c^b = \Pi_b(b^* c)$.

We have proved that if a lex category $\mathcal{C}$ has dependent product $\Pi_f : \mathcal{C}/A \to \mathcal{C}/B$ along every morphism $f : A \to B$ then it is locally cartesian closed. The converse holds as well, that is every lccc has dependent products. For a proof see [MM92, Theorem I.9.4].

Exercise 4.4.9 In Set consider the dependent sum and product along a function $!_I : I \to 1$. Show that for $a : A \to I$ the set $\Pi_{!_I} A$ is the set of right inverses of a:

$$\Pi_{!_I} A = \left\{ s : I \to A \;\middle|\; a \circ s = 1_I \right\} \ .$$

If $(A_i)_{i\in I}$ is a family of sets indexed by I and we take

$$A = \coprod_{i\in I} A_i = \left\{ \langle i, x\rangle \in I \times \bigcup_{i\in I} A_i \;\middle|\; i \in I \wedge x \in A_i \right\}$$

with $a = \pi_0 : \langle i, x\rangle \mapsto i$ then $\Pi_{!_I} A$ is precisely the cartesian product $\Pi_{i\in I} A_i$. Calculate what Π_f is in Set for a general $f : J \to I$ and conclude that Set is locally cartesian closed.

Proposition 4.4.10 *In an lccc $\mathcal{C}$, for any $f : A \to B$ the change of base functor $f^* : \mathcal{C}/B \to \mathcal{C}/A$ preserves the ccc structure.*

Proof. We need to show that f^* preserves terminal objects, binary products, and exponentials in slices. Because f^* is a right adjoint it preserves limits, hence it preserves terminal objects and binary products. To see that it preserves exponentials we first show that $f^* \circ \Pi_g \cong \Pi_{f^*g} \circ (g^* f)^*$ for $g : C \to B$. Given any $d : D \to C$, and $e : E \to A$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{e \to f^*(\Pi_g d)}{\Sigma_f e \to \Pi_g d}}{g^*(\Sigma_f e) \to d}}{g^*(f \circ e) \to d}}{(g^* f) \circ ((f^* g)^* e) \to d}}{(f^* g)^* e \to (g^* f)^* d}}{e \to \Pi_{f^*g}((g^* f)^* d)}$$

By Yoneda Lemma it follows that $f^*(\Pi_g d) \cong \Pi_{f^*g}((g^* f)^* d)$. Now we have, for any $d : D \to A$ and $c : C \to A$,

$$f^* c^d = f^*(\Pi_d(d^* c)) = \Pi_{f^*d}((d^* f)^*(d^* c)) = \Pi_{f^*d}((f^* d)^*(f^* c)) = (f^* c)^{(f^* d)} .$$

■

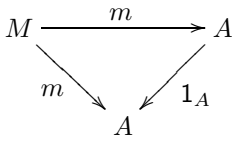
Exercise 4.4.11 State precisely which instance of Yoneda Lemma is used in the conclusion of the last proof.

Exercise 4.4.12 In the preceding proof we used the fact that $(d^* f)^*(d^* c) \cong (f^* d)^*(f^* c)$ and $g^*(f \circ e) \cong (g^* f) \circ ((f^* g)^* e)$. Prove that this is really so.

Locally cartesian closed categories are an important example of categories with universal quantifiers.

Proposition 4.4.13 *A locally cartesian closed category has universal quantifiers.*

Proof. Suppose $\mathcal{C}$ is locally cartesian closed. First observe that a morphism $m : M \to A$ is mono if, and only if, the morphism



is mono in $\mathcal{C}/A$. Because right adjoints preserve monos, $\Pi_f : \mathcal{C}/A \to \mathcal{C}/B$ preserve monos for any $f : A \to B$, that is, if $m : M \rightarrowtail A$ is mono then $\Pi_f m : \Pi_f M \to B$ is mono in $\mathcal{C}$. Therefore, we may define $\forall_f$ as the restriction of Π_f to $\mathsf{Sub}(A)$. To be more precise, a subobject $[m : M \rightarrowtail A]$ is mapped by $\forall_f$ to the subobject $[\Pi_f m : \Pi_f M \rightarrowtail B]$. This works because for any monos $m : M \rightarrowtail A$ and $n : N \rightarrowtail B$ we have

$$\begin{array}{rr} f^*[m : M \rightarrowtail A] \leq [n : N \rightarrowtail B] & \text{in } \mathsf{Sub}(B) \\ \hline\hline f^*m \to n & \text{in } \mathcal{C}/B \\ \hline\hline m \to \Pi_f n & \text{in } \mathcal{C}/A \\ \hline\hline [m] \leq \forall_f[n] & \text{in } \mathsf{Sub}(A) \end{array}$$

The Beck-Chevalley condition for $\forall_f$ follows from Proposition 4.4.10. Indeed, if $g : C \to B$ and $m : M \rightarrowtail C$ then

$$f^*(\Pi_g m) \cong \Pi_{f^*g}((g^*f)^*m) \ ,$$

therefore

$$f^*(\forall_g[m : M \rightarrowtail C]) = \forall_{f^*g}((g^*f)^*[m : M \rightarrowtail C]) \ ,$$

as required. ∎

4.4.3 Implication from Universal Quantifiers

Recall that the rules of inference for implication state that $\Rightarrow$ is right adjoint to $\wedge$:

$$\frac{\Gamma \mid \theta \ \texttt{pred} \qquad \Gamma \mid \varphi \ \texttt{pred}}{\Gamma \mid (\theta \Rightarrow \varphi) \ \texttt{pred}} \qquad\qquad \frac{\Gamma \mid \Psi, \theta \vdash \varphi}{\Gamma \mid \Psi \vdash \theta \Rightarrow \varphi}$$

Exercise 4.4.14 Show that the above two-way rule can be replaced by the following introduction and elimination rules:

$$\frac{\Gamma \mid \Psi\theta \vdash \varphi}{\Gamma \mid \Psi \vdash \theta \Rightarrow \varphi} \qquad\qquad \frac{\Gamma \mid \Psi \vdash \theta \Rightarrow \varphi \qquad \Gamma \mid \Psi \vdash \theta}{\Gamma \mid \Psi \vdash \varphi}$$

We expect that in order to interpret implication in a lex category $\mathcal{C}$ we require $\mathsf{Sub}(A)$ to be a Heyting algebra for every $A \in \mathcal{C}$. However, we must not forget that implication interacts with substitution by the rule

$$(\theta \Rightarrow \varphi)[t/x] \;=\; \theta[t/x] \Rightarrow \varphi[t/x] \; .$$

Semantically this means that implication is *stable* under pullbacks.

Definition 4.4.15 A lex category $\mathcal{C}$ has *implications* when, for every $A \in \mathcal{C}$, the poset $\mathsf{Sub}(A)$ is a Heyting algebra with stable implication $\Rightarrow$. This means that for $U, V \in \mathsf{Sub}(A)$ and $f : B \to A$,

$$f^*(U \Rightarrow V) = (f^*U \Rightarrow f^*V) \; .$$

Proposition 4.4.16 *If a lex category has universal quantifiers then it has implications.*

Proof. Let $[u : U \rightarrowtail A]$ and $[v : V \rightarrowtail A]$ be subobjects of A. Define

$$([u] \Rightarrow [v]) = \forall_u(u^*[v]) \; .$$

Then for any subobject $[w : W \rightarrowtail A]$

$$\begin{array}{rr}
[w] \leq [u] \Rightarrow [v] & \text{in } \mathsf{Sub}(A) \\
\hline\hline
[w] \leq \forall_u(u^*[v]) & \text{in } \mathsf{Sub}(A) \\
\hline\hline
u^*[w] \leq u^*[v] & \text{in } \mathsf{Sub}(U) \\
\hline\hline
u^*w \to u^*v & \text{in } \mathcal{C}/U \\
\hline\hline
\Sigma_u(u^*w) \to v & \text{in } \mathcal{C}/A \\
\hline\hline
u \circ (u^*w) \to v & \text{in } \mathcal{C}/A \\
\hline\hline
[u] \wedge [w] \leq [v] & \text{in } \mathsf{Sub}(A)
\end{array}$$

Stability of $\Rightarrow$ follows from Beck-Chevalley condition for $\forall$. ∎

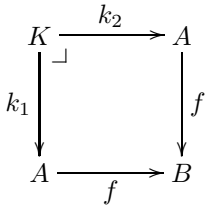
Exercise 4.4.17 Prove the last claim of the proof.

4.5 Regular Logic

In this section we consider the question when a lex category has existential quantifiers. It turns out that this is related to the notion of a *regular category*, which was arose independently of categorical logic.

4.5.1 Regular Categories

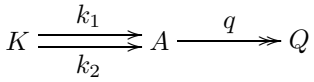
Throughout this section we work in a lex category $\mathcal{C}$. The *kernel pair* of a morphism $f : A \to B$ is the pair of morphisms $k_1, k_2 : K \to A$ obtained as in the following pullback



Note that a kernel pair determines an equivalence relation $\langle k_1, k_2 \rangle : K \rightarrowtail A \times A$. Without going into details about what an equivalence relation is a general lex category, let us just note that in Set the mono $\langle k_1, k_2 \rangle : K \rightarrowtail A \times A$ corresponds to the equivalence relation $\sim$ on A defined by

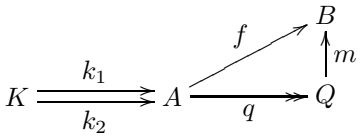
$$x \sim y \iff fx = fy \ .$$

The quotient by the equivalence relation determined by the kernel pair k_1, k_2 is their coequalizer $q : A \to Q$, if it exists,



Such a coequalizer is called a *kernel quotient*.

Because $f \circ k_1 = f \circ k_2$, f factors through q by a unique morphism $m : Q \to A$,



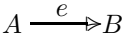
(4.5)

It is of some interest to know when m is guaranteed to be a mono. For example, in Set the function $m : Q \to B$ is defined by $m[x] = fx$, where $Q = A/\!\sim$ as above. In this case m is injective because $m[x] = m[y]$ implies $fx = fy$, hence $x \sim y$ and $[x] = [y]$.

Definition 4.5.1 A category with finite limits is *regular* when it has kernel quotients and stable regular epis, meaning that:

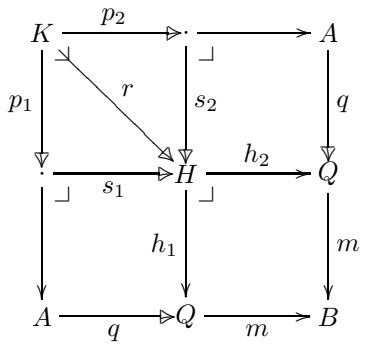
1. every kernel pair has a coequalizer, and
2. a pullback of a regular epi is a regular epi.

Recall that an epi is *regular* if it is a coequalizer. In diagrams we denote regular epis by arrows with triangular heads, for example



Exercise 4.5.2 Suppose $e : A \twoheadrightarrow B$ is a regular epi. Prove that it is the coequalizer of its kernel pair.

Let us return to (4.5) and show that in a regular category m is mono. Consider the following diagram, in which h_1, h_2 are constructed as the kernel pair of m, and the other three squares are constructed as pullbacks:

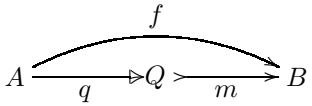


Because all the smaller squares are pullbacks the large square is a pullback as well, therefore the morphism across the top is $k_2 : K \to A$, and the left-hand vertical morphism is $k_1 : K \to A$. Morphisms s_1, s_2, p_1, and p_2 are all regular epis because they are pullbacks of regular epis. The morphism $r = s_2 \circ p_2 = s_1 \circ p_1$ is epi because it is a composition of regular epis. Observe that

$$h_1 \circ r = q \circ k_1 = q \circ k_2 = h_2 \circ r \ ,$$

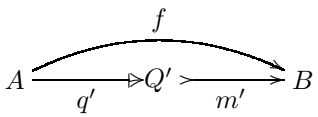
and because r is epi, $h_1 = h_2$. But this means that m is monic, since given $u, v : U \to Q$ with $m \circ u = m \circ v$ there exists $w : U \to H$ such that $u = w \circ h_1 = w \circ h_2 = v$.

Proposition 4.5.3 *In a regular category every morphism $f : A \to B$ factors as a composition of a regular epi q and a mono m,*



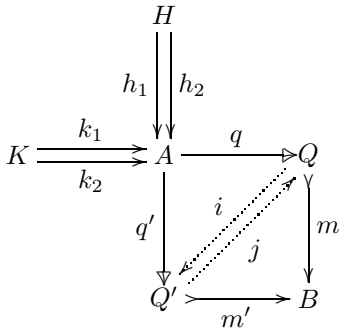
The factorization in unique up to isomorphism.

Proof. By uniqueness of factorization we mean that if



is another such factorization, then there exists an isomorphism $i : Q \to Q'$ such that $q' = i \circ q$ and $m = m' \circ m$.

As the factorization of f we take the one constructed in (4.5). Then q is regular epi by construction, and we have just shown that m is mono. So it only remains to show that the factorization is unique. Suppose f also factors as $f = m' \circ q'$ where q' is regular epi and m is mono. Consider the following diagram, in which k_1, k_2 is the kernel pair of f, q is the coequalizer of k_1 and k_2, and h_1, h_2 is the kernel pair of q' so that q' is the coequalizer of h_1 and h_2:



Because $m' \circ q' \circ k_1 = m \circ q \circ k_1 = m \circ q \circ k_2 = m' \circ q' \circ k_2$ and m' is mono, $q' \circ k_1 = q' \circ k_2$. So there exists a unique $i : Q \to Q'$ such that $q' = i \circ q$. But then $m' \circ i \circ q = m' \circ q' = f = m \circ q$ and because q is epi, $m' \circ i = m$.

We prove that i is iso by constructing its inverse j. Because $m \circ q \circ h_1 = m' \circ q \circ h_1 = m' \circ q \circ h_2 = m \circ q \circ h_2$ and m is mono, $q \circ h_1 = q \circ h_2$. So there exists a unique $j : Q' \to Q$ such that $q = j \circ q'$. Now we have $i \circ j \circ q' = i \circ q = \mathsf{1}_{Q'} \circ q'$, from which we conclude that $i \circ j = \mathsf{1}_{Q'}$ because q' is epi. Similarly, $j \circ i \circ q = j \circ q' = \mathsf{1}_Q \circ q$, therefore $j \circ i = \mathsf{1}_Q$. ∎

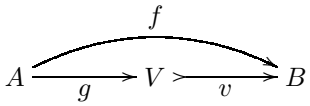
A factorization $f = m \circ q$ as in the previous Proposition determines a subobject

$$\mathsf{im}(f) = [m : Q \rightarrowtail B] \in \mathsf{Sub}(B) \ ,$$

called the *image of* f. It is characterized as the least subobject $[u : U \rightarrowtail B]$ of B through which f factors.

Proposition 4.5.4 *In a regular category $\mathcal{C}$, the image $\mathsf{im}(f)$ of a morphism $f : A \to B$ is the least subobject $[u : U \rightarrowtail B]$ of B such that f factors through u.*

Proof. Suppose f factors through $v : V \rightarrowtail B$ as



and consider the factorization of f, as in (4.5). Since $v \circ g \circ k_1 = f \circ k_1 = f \circ k_2 = v \circ g \circ k_2$ and v is mono, $g \circ k_1 = g \circ k_2$, therefore there exists a unique $\overline{g} : Q \to V$ such that $g = \overline{g} \circ q$. Now $v \circ \overline{g} \circ q = v \circ g = f = m \circ q$ and because q is epi, $v \circ \overline{g} = m$ as required. ∎

A morphism $f : A \to B$ is sometimes called a *generalized element* or *generalized point of* B. If $U \leq B$ is a subobject of B, we write $f \in U$ when f factors through U. With this notation we have

$$f \in U \iff \mathsf{im}(f) \leq U \ .$$

At first sight it may seem unreasonable to use the phrases "element" and "point" for something that clearly is a morphism. But this is in perfect accordance with mathematical practice. For example, in mechanics we often say things like "the velocity $\dot{p}$ of a point mass p moving in space" which just means that p is a vector in $\mathbb{R}^3$ parametrized by time—in other words a generalized point $p : \mathbb{R} \to \mathbb{R}^3$.

In set theory the axiom of extensionality say that "a set is determined by its elements", that is, for any sets A and B,

$$A = B \iff \forall\, x\,.\,(x \in A \iff x \in B) \ .$$

The corresponding statement in an arbitrary category is that objects A and B are isomorphic if, and only if, they have naturally isomorphic sets of *generalized* elements:

$$A \cong B \iff \mathsf{Hom}(-, A) \cong \mathsf{Hom}(-, B) \ .$$

This is Corollary 1.4.5, which says that the Yoneda embedding reflects isomorphisms.

Let us consider some examples of regular categories. The category Set is regular. It is complete and cocomplete, so it has finite limits and coequalizers. The pullback of a regular epi is again regular epi because in Set every epi is regular, and it is always the case that the pullback of an epi is epi.

In general, any presheaf category $\widehat{\mathcal{C}}$ is regular because it is complete and cocomplete, and every epi is regular.

The next example deserves to be a proposition.

Proposition 4.5.5 *The category* $\mathsf{Mod}_{\mathsf{Set}}(\mathbb{A})$ *of set-theoretic models of an algebraic theory* $\mathbb{A}$ *is regular.*

Proof. We sketch a proof, for details see [Bor94b, Theorem 3.5.4]. Recall that the objects of $\mathsf{Mod}(\mathbb{A}) = \mathsf{Mod}_{\mathsf{Set}}(\mathbb{A})$ are $\mathbb{A}$-algebras, which are structures $X = (|X|, f_1, f_2, \ldots)$ where $|X|$ is the carrier set and $f_1, f_2, \ldots$ are the basic operations on $|X|$. Every such $\mathbb{A}$-algebra is also required to satisfy the equational axioms of $\mathbb{A}$. A morphism $f : X \to Y$ is a function $f : |X| \to |Y|$ that preserves the basic operations.

The category of $\mathbb{A}$-algebras $\mathsf{Mod}(\mathbb{A})$ has small limits, which are computed as in Set. Thus the product of $\mathbb{A}$-algebras X and Y has as its carrier set $|X \times Y| = |X| \times |Y|$, and the basic operations of $X \times Y$ are computed separately on each component. An equalizer of morphisms $f, g : X \to Y$ has as its carrier set the equalizer of $f, g : |X| \to |Y|$, and the basic operations inherited from X.

To see that coequalizers of kernel pairs exist, consider a morphism $h : X \to Y$. We can form the quotient $\mathbb{A}$-algebra Q whose carrier set is $|Q| = |X|/\!\sim$, where $\sim$ is the relation defined by

$$x \sim y \iff hx = hy \ .$$

A basic operation $f_Q : Q^k \to Q$ is induced by the basic operation $f_X : X^k \to X$ by

$$f_Q\langle [x_1], \ldots, [x_k]\rangle = [f_X\langle x_1, \ldots, x_k\rangle] \ .$$

It is easily verified that Q is an $\mathbb{A}$-algebra and that the canonical quotient map $q : |X| \to |Q|$ is the coequalizer of the kernel pair of h.

Lastly regular epis in $\mathsf{Mod}(\mathbb{A})$ are stable because pullbacks and kernel pairs are computed as in Set, and a morphism $f : X \to Y$ is regular epi in $\mathsf{Mod}(\mathbb{A})$ if, and only if, the underlying function $f : |X| \to |Y|$ is regular epi in Set. ∎

We now know that categories of groups, rings, modules, $\mathcal{C}^\infty$-rings and other algebraic categories are regular. The preceding proposition is useful also for showing that certain structures cannot be axiomatized by algebraic theories. For example the category of posets is not regular, therefore the theory of partial orders cannot be axiomatized solely by equations.

Exercise 4.5.6 Why is Poset not regular?

Exercise* 4.5.7 Is Top regular? Hint: is there is a topological quotient map $q : X \twoheadrightarrow X'$ and a space Y such that $q \times 1_Z : X \times Y \twoheadrightarrow X' \times Y$ is not a quotient map?

Definition 4.5.8 A functor $F : \mathcal{C} \to \mathcal{D}$ is *regular* if it preserves finite limits and regular epis. It follows that F preserves image factorizations. The category of regular functors $\mathcal{C} \to \mathcal{D}$ and natural transformations is denoted by $\mathsf{Reg}(\mathcal{C}, \mathcal{D})$.

Exercise 4.5.9 Let H be a complete Heyting algebra. The category of *H-sets* has as its objects $X = (|X|, e_X : |X| \to H)$ where $|X|$ is a set and e_X is a function, called the *existence predicate of X*. For $x \in |X|$, $e_X x$ is the "amount by which x exists as an element of $|X|$". A morphism $f : X \to Y$ is a function $f : |X| \to |Y|$ satisfying, for all $x \in |X|$,

$$e_X x \leq e_Y(fx) \ .$$

Prove that H-sets form a regular category.

4.5.2 Images and existential quantifiers

Recall that $\mathsf{Sub}(A)$ is equivalent to the category $\mathsf{Mono}(A)$ of monos into A. If we compose an equivalence functor $\mathsf{Sub}(A) \to \mathsf{Mono}(A)$ with the inclusion $\mathsf{Mono}(A) \to \mathcal{C}/A$ we obtain an inclusion functor $I : \mathsf{Sub}(A) \to \mathcal{C}/A$. In the other direction we have the "image functor" $\mathsf{im} : \mathcal{C}/A \to \mathsf{Sub}(A)$ which maps an object $b : B \to A$ in $\mathcal{C}/A$ to the subobject $\mathsf{im}(f) = [\mathsf{im}(f) \rightarrowtail A]$.

Exercise 4.5.10 In order to show that im is in fact a functor, prove that $f = g \circ h$ implies $\mathsf{im}(f) \leq \mathsf{im}(g)$.

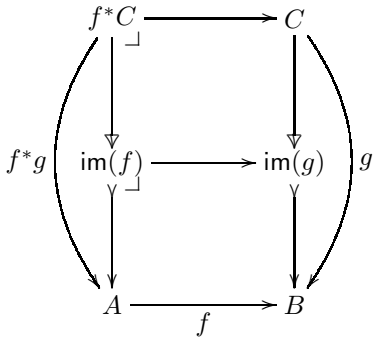
Proposition 4.5.4 says that the image functor is left adjoint to the inclusion functor $I : \mathsf{Sub}(A) \to \mathcal{C}/A$,

$$\mathsf{im} \dashv I \ .$$

Furthermore, images are stable in the sense that the following diagram commutes for all $f : A \to B$:

$$\begin{array}{ccc} \mathcal{C}/A & \xleftarrow{f^*} & \mathcal{C}/B \\ {\scriptstyle \mathsf{im}_A}\downarrow & & \downarrow{\scriptstyle \mathsf{im}_B} \\ \mathsf{Sub}(A) & \xleftarrow[f^*]{} & \mathsf{Sub}(B) \end{array} \tag{4.6}$$

The functor f^* on the top is the change of base functor, and the functor f^* on the bottom is the pullback functor. To see this, consider $g : C \to B$ and the following diagram:



On the right-hand side we have the factorization of g, which is then pulled back along f. Because monos and regular epis are stable, this gives us a factorization of f^*g, hence

$$\mathsf{im}(f^*g) = f^*(\mathsf{im}(g)) \ .$$

Proposition 4.5.11 *A regular category has existential quantifiers. The existential quantifier along $f : A \to B$ is*

$$\exists_f [m : M \rightarrowtail A] = \mathsf{im}(f \circ m) \ .$$

Proof. First we verify that $\exists_f \dashv f^*$. For subobjects $[u : U \rightarrowtail A]$ and $[v : V \rightarrowtail B]$:

$\exists_f[u] \leq [v]$	in $\mathsf{Sub}(B)$
$\mathsf{im}(f \circ u) \leq [v]$	in $\mathsf{Sub}(B)$
$f \circ u \to I[v]$	in $\mathcal{C}/B$
$f \circ u \to v$	in $\mathcal{C}/B$
$\Sigma_f u \to v$	in $\mathcal{C}/B$
$u \to f^* v$	in $\mathcal{C}/A$
$[u] \leq f^*[v]$	in $\mathsf{Sub}(A)$

In the second step in the above derivation we used the adjunction between $\mathsf{im} : \mathcal{C}/B \to \mathsf{Sub}(B)$ and the inclusion $\mathsf{Sub}(B) \to \mathcal{C}/B$.

The Beck-Chevalley condition follows from stability of image factorizations. Given a pullback

$$\begin{array}{ccc} D & \xrightarrow{h} & C \\ {\scriptstyle k}\downarrow & \lrcorner & \downarrow{\scriptstyle g} \\ A & \xrightarrow[f]{} & B \end{array}$$

and a mono $u : U \rightarrowtail C$, (4.6) gives

$$f^*(\exists_g[u]) = f^*(\mathsf{im}(g \circ u)) = \mathsf{im}(f^*(g \circ u)) = \mathsf{im}(k \circ (h^* u)) = \exists_k(h^* u) = \exists_k(h^*[u]) \,.$$

■

4.5.3 Regular Theories

A regular category has finite limits and image factorizations, therefore it allows us to interpret a type theory with the terminal type and binary products, and a logic with equality, conjunction, and existential quantifiers. Such logic is called *regular logic*.

Definition 4.5.12 A multi-sorted *regular theory* $\mathbb{T}$ is a multi-sorted type theory together with axioms expressed in the fragment of logic built from $=$, $\wedge$, and $\exists$.

Let us recall that an *interpretation* of a regular theory $\mathbb{T}$ in a regular category $\mathcal{C}$ is given, as usual, by the following data:

1. Each basic sort A is interpreted as an object $[\![A]\!]$.

2. Each basic constant f with signature $(A_1, \ldots, A_n; B)$ is interpreted as a morphism $[\![f]\!] : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![B]\!]$.

3. Each basic relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted as a subobject $[\![R]\!] \in \mathsf{Sub}([\![A_1]\!] \times \cdots \times [\![A_1]\!])$.

An interpretation is then extended to all terms and formulas of the theory $\mathbb{T}$. The lex part of logic is interpreted as was explained in Section 4.3. Existential quantification is interpreted by the existentials in the category,

$$[\![\Gamma \mid \exists\, x{:}A\,.\,\varphi]\!] = \exists_A [\![\Gamma, x{:}A \mid \varphi]\!] \ ,$$

where $\exists_A = \exists_{\pi_0}^{\Gamma, A} : \mathsf{Sub}([\![\Gamma]\!] \times [\![A]\!]) \to \mathsf{Sub}([\![\Gamma]\!])$.

If all the axioms of $\mathbb{T}$ are valid in a given interpretation, then we say that the interpretation is a *model* of $\mathbb{T}$. Once again we would like to show that semantics of regular categories is *functorial*, i.e., that the models of a theory $\mathbb{T}$ can be viewed as structure-preserving functors,

$$\mathsf{Reg}(\mathcal{S}(\mathbb{T}), \mathcal{C}) \simeq \mathsf{Mod}_{\mathcal{C}}(\mathbb{T}) \ .$$

where on the left-hand side $\mathcal{S}(\mathbb{T})$ is a suitable regular category, called the *classifying category for* $\mathbb{T}$, and $\mathsf{Reg}(-,-)$ indicates that we take regular functors. Once we do this, we can also define morphisms between models to be natural transformations.

Let us sketch the construction of $\mathcal{S}(\mathbb{T})$. An object is represented by a formula in a context

$$[\Gamma \mid \varphi]$$

where $\Gamma \mid \varphi$ `pred`. Two such objects $[\Gamma \mid \varphi]$ and $[\Gamma \mid \psi]$ are equal if $\mathbb{T}$ proves both

$$\Gamma \mid \varphi \vdash \psi \ , \qquad\qquad \Gamma \mid \psi \vdash \varphi \ .$$

Objects which differ only in the names of free variables are also considered equal. A morphism

$$[x{:}A \mid \varphi] \xrightarrow{\ \rho\ } [y{:}B \mid \psi]$$

is represented by a formula $x{:}A, y{:}B \mid \rho$ such that $\mathbb{T}$ proves that ρ is a *functional relation* from φ to ψ:

$$\begin{aligned} x{:}A \mid \varphi &\vdash \exists\, y{:}B\,.\,\rho && \text{(total)} \\ x{:}A, y{:}B, z{:}B \mid \rho, \rho[z/y] &\vdash y = z && \text{(single-valued)} \\ x{:}A, y{:}B \mid \rho &\vdash \varphi \wedge \psi && \text{(well-defined)} \end{aligned}$$

Two functional relations ρ and σ represent the same morphism if $\mathbb{T}$ proves both

$$x{:}A, y{:}B \mid \rho \vdash \sigma \ , \qquad\qquad x{:}A, y{:}B \mid \sigma \vdash \rho \ ,$$

Relations which only differ in the names of free variables are considered equal.

The identity morphism on $[x{:}A \mid \varphi]$ is represented by the relation

$$x{:}A, y{:}A \mid (x = y) \wedge \varphi \wedge \varphi[y/x] \ .$$

Composition of morphisms

$$[x{:}A \mid \varphi] \xrightarrow{\ \rho\ } [y{:}B \mid \psi] \xrightarrow{\ \tau\ } [z{:}C \mid \theta]$$

is given by the relation

$$x{:}A, z{:}C \mid \exists\, y{:}B \,.\, (\rho \wedge \tau) \ .$$

We leave a detailed proof that $\mathcal{S}(\mathbb{T})$ as exercise. Let only show that composition of morphisms is associative. Given morphisms

$$[x{:}A \mid \varphi] \xrightarrow{\ \rho\ } [y{:}B \mid \psi] \xrightarrow{\ \tau\ } [z{:}C \mid \theta] \xrightarrow{\ \sigma\ } [t{:}D \mid \zeta]$$

we need to derive in context $x{:}A, t{:}D$

$$\xi \ = \ \exists\, y{:}B \,.\, (\varphi \wedge (\exists\, z{:}C \,.\, (\tau \wedge \sigma)))$$

from

$$\exists\, z{:}C \,.\, ((\exists\, y{:}B \,.\, (\rho \wedge \tau)) \wedge \sigma)$$

and vice versa. In one direction we have:

$$\frac{\frac{\frac{\frac{\frac{\frac{\frac{\frac{x{:}A, t{:}D \mid \exists\, y{:}B \,.\, (\rho \wedge \exists\, z{:}C \,.\, (\tau \wedge \sigma)) \vdash \xi}{x{:}A, t{:}D, y{:}B \mid \rho \wedge \exists\, z{:}C \,.\, (\tau \wedge \sigma) \vdash \xi}}{x{:}A, t{:}D, y{:}B \mid \rho, \exists\, z{:}C \,.\, (\tau \wedge \sigma) \vdash \xi}}{x{:}A, t{:}D, z{:}C, y{:}B \mid \rho, \tau \wedge \sigma \vdash \xi}}{x{:}A, t{:}D, z{:}C, y{:}B \mid \rho, \tau, \sigma \vdash \xi}}{x{:}A, t{:}D, z{:}C, y{:}B \mid \rho \wedge \tau, \sigma \vdash \xi}}{x{:}A, t{:}D, z{:}C \mid (\exists\, y{:}B \,.\, (\rho \wedge \tau)), \sigma \vdash \xi}}{x{:}A, t{:}D, z{:}C \mid (\exists\, y{:}B \,.\, (\rho \wedge \tau)) \wedge \sigma \vdash \xi}}{x{:}A, t{:}D \mid \exists\, z{:}C \,.\, ((\exists\, y{:}B \,.\, (\rho \wedge \tau)) \wedge \sigma) \vdash \xi}$$

The other direction is equally boring.

Exercise 4.5.13 Extend the definition of $\mathcal{S}(\mathbb{T})$ to morphisms between objects with arbitrary contexts,

$$[\Gamma \mid \varphi] \xrightarrow{\ \rho\ } [\Delta \mid \psi]$$

and provide a proof that $\mathcal{S}(\mathbb{T})$ is a category.

The category $\mathcal{S}(\mathbb{T})$ is a regular. We sketch the constructions required for regularity. The terminal object is $[\cdot \mid \top]$. The product of $[x{:}A \mid \varphi]$ and $[y{:}B \mid \psi]$, where x and y are distinct variables, is the object

$$[x'{:}A, y'{:}B \mid \varphi[x'/x] \wedge \psi[y'/y]] \,.$$

The first projection from the product is

$$x'{:}A, y'{:}B, x{:}A \mid x' = x \wedge \varphi[x'/x] \wedge \psi[y'/y]$$

and the second projection is

$$x{:}A, y{:}B, y'{:}B \mid y' = y' \wedge \varphi[x'/x] \wedge \psi[y'/y] \,.$$

An equalizer of morphisms

$$[x{:}A \mid \varphi] \underset{\tau}{\overset{\rho}{\rightrightarrows}} [y{:}B \mid \psi]$$

is

$$[x{:}A \mid \varphi \wedge \exists\, y{:}B \,.\, (\rho \wedge \tau)] \xrightarrow{\quad\varepsilon\quad} [x'{:}A \mid \varphi[x'/x]]$$

where ε is the morphism

$$x{:}A, x'{:}A \mid (x = x') \wedge \varphi \wedge \exists\, y{:}B \,.\, (\rho \wedge \tau) \;\;.$$

Finally, let us consider coequalizers of kernel pairs. The kernel pair of $\rho : [x{:}A \mid \varphi] \to [y{:}B \mid \psi]$ is

$$K \underset{\kappa_2}{\overset{\kappa_1}{\rightrightarrows}} [x{:}A \mid \varphi]$$

where K is the object

$$[u{:}A, v{:}A \mid \varphi[u/x] \wedge \varphi[v/x] \wedge \exists\, y{:}B \,.\, (\rho[u/x] \wedge \rho[v/x])] \,,$$

the morphism κ_1 is

$$u{:}A, v{:}A, x{:}A \mid (u = x) \wedge \varphi$$

and κ_2 is

$$u{:}A, v{:}A, x{:}A \mid (v = x) \wedge \varphi \,.$$

Now the coequalizer of κ_1 and κ_2 is the morphism

$$[x{:}A \mid \varphi] \xrightarrow{\quad\rho\quad} [y{:}B \mid \exists\, x{:}A \,.\, \rho] \;\;.$$

Exercise 4.5.14 Show that in $\mathcal{S}(\mathbb{T})$ the regular-epi mono factorization of a morphism $\rho : [x{:}A \mid \varphi] \to [y{:}B \mid \psi]$ is given by

$$[x{:}A \mid \varphi] \xrightarrow{\quad\rho\quad} [y{:}B \mid \exists\, x{:}A \,.\, \rho] \xrightarrow{\quad\iota\quad} [z{:}B \mid \psi[z/y]]$$

where ι is the morphism

$$y{:}B, z{:}B \mid (y = z) \wedge (\exists\, x{:}A \,.\, \rho) \wedge \psi[z/y] \,.$$

Theorem 4.5.15 *Semantics of regular logic in regular categories is sound and complete: a regular theory $\mathbb{T}$ proves*

$$\Gamma \mid \varphi \vdash \psi$$

if, and only if, every model of $\mathbb{T}$ satisfies it.

Proof. Soundness is proved by induction on the proof of $\Gamma \mid \varphi \vdash \psi$. All that needs to be checked is that rules of inference preserve validity. This was done for lex logic in Section 4.3. The inference rule for existential quantifiers preserves validity as well, because it translates to the universal property of the adjunction $\exists \dashv \pi_0$.

Completeness follows from the fact that $\mathbb{T}$ has a universal model U in $\mathcal{S}(\mathbb{T})$. In this model a sort A is interpreted by the object $[x{:}A \mid \top]$ and a basic constant f with signature $(A_1, \ldots, A_n; B)$ is interpreted by the relation

$$x_1{:}A_1, \ldots, x_n{:}A_n, y{:}B \mid f(x_1, \ldots, x_n) = y \ .$$

A relation symbol R with signature $(A_1, \ldots, A_n)$ is interpreted by the subobject represented by the morphism

$$\rho : [x_1{:}A_1, \ldots, x_n{:}A_n \mid R(x_1, \ldots, x_n)] \longrightarrow [y_1{:}A_1, \ldots, y_n{:}A_n \mid \top]$$

where ρ is the formula

$$R(x_1, \ldots, x_n) \wedge x_1 = y_1 \wedge \cdots \wedge x_n = y_n \ .$$

The model U has the property

$$U \models \ \Gamma \mid \varphi \vdash \psi \quad \iff \quad \mathbb{T} \text{ proves } \Gamma \mid \varphi \vdash \psi \ .$$

■

4.6 First-order Logic

Bibliography

[Bor94a] F. Borceux. *Handbook of Categorical Algebra I. Basic Category Theory*, volume 51 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1994.

[Bor94b] F. Borceux. *Handbook of Categorical Algebra II. Categories and Structures*, volume 51 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 1994.

[Joh82] P.T. Johnstone. *Stone Spaces*. Number 3 in Cambridge studies in advanced mathematics. Cambridge University Press, 1982.

[McC93] W. McCune. Single axioms for groups and abelian groups with various operations. *Journal of Automated Reasoning*, 10(1):1–13, 1993.

[MM92] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory*. Springer-Verlag, New York, 1992.

Index