Introduction to Categorical Logic

[DRAFT: May 1, 2022]

Steve Awodey Andrej Bauer

Contents

Chapter 3

Cartesian Closed Categories and the λ-Calculus

3.1 Categorification and the Curry-Howard correspondence

Consider the following natural deduction proof in propositional calculus.

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

This deduction shows that

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

But so does the following:

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

As does:

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

There is a sense in which the first two proofs are "equivalent", but not the first and the third. The relation (or property) of *provability* in propositional calculus $\vdash \phi$ discards such differences in the proofs that witness it. According to the "proof-relevant" point of view, sometimes called *propositions as types*, one retains as relevant some information about the way in which a proposition is proved. This is effected by annotating the proofs with *proof-terms* as they are constructed, as follows:

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

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

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

The proof terms for the first two proofs are the same, namely $\lambda x.\pi_2(x)(\pi_1(\pi_1(x)))$, but the term for the third one is $\lambda x.\pi_2(\pi_1(x))$, reflecting the difference in the proofs. The assignment works by labelling assumptions as variables, and then associating term-constructors to the different rules of inference: pairing and projection to conjunction introduction and elimination, function application and λ-abstraction to implication elimination (*modus ponens*) and introduction. The use of variable binding to represent cancellation of premisses is a particularly effective device.

From the categorical point of view, the relation of deducibility $\phi \vdash \psi$ is a mere preorder. The addition of proof terms $x : \phi \vdash t : \psi$ results in a *categorification* of this preorder, in the sense that it is a "proper" category, the preordered reflection of which is the deducibility preorder. And now the following remarkable fact emerges: it is hardly surprising that the deducibility preorder has, say, finite products $\phi \wedge \psi$ or even exponentials $\phi \Rightarrow \psi$; but it is *amazing* that the category with proof terms $x : \phi \vdash t : \psi$ as arrows, also turns out to be a cartesian closed category, and indeed a proper one, with distinct parallel arrows, such as

$$\begin{aligned} \pi_2(x)(\pi_1(\pi_1(x))) : (A \wedge B) \wedge (A \Rightarrow B) &\longrightarrow B, \\ \pi_2(\pi_1(x)) : (A \wedge B) \wedge (A \Rightarrow B) &\longrightarrow B. \end{aligned}$$

This *category of proofs* contains information about the “proof theory” of the propositional calculus, as opposed to its mere relation of deducibility. The calculus of proof terms can be presented formally in a system of *simple type theory*, with an alternate interpretation as a formal system of function application and abstraction. This dual interpretation—as the proof theory of propositional logic, and as a system of type theory for the specification of functions—is called the *Curry-Howard correspondence* []. From the categorical point of view, it expresses the structural equivalence between the cartesian closed categories of proofs in propositional logic and terms in simple type theory. Both of these can be seen as categorifications of their preorder reflection, the deducibility preorder of propositional logic (cf. [MH92]).

In the following sections, we shall consider this remarkable correspondence in detail, as well as some extensions of the basic case represented by cartesian closed categories: categories with coproducts, cocomplete categories, and categories equipped with modal operators. In the next chapter, it will be seen that this correspondence even extends to proofs in quantified predicate logic and terms in dependent type theory, and beyond.

3.2 Cartesian closed categories

Exponentials

We begin with the notion of an exponential B^A of two objects A, B in a category, motivated by a couple of important examples. Consider first the category Pos of posets and monotone functions. For posets P and Q the set $\mathsf{Hom}(P,Q)$ of all monotone functions between them is again a poset, with the pointwise order:

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

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

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

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

Thus the category Mon also admits such “internal Hom”s. The same thing works in the category Group of groups and group homomophisms, where the set $\mathsf{Hom}(G,H)$ of all homomorphisms between groups G and H can be given a pointwise group structure.

These examples suggest a general notion of “internal Hom” in a category: an “object of morphisms $A \to B$” which corresponds to the hom-set $\mathsf{Hom}(A,B)$. The other ingredient needed is an “evaluation” operation $\epsilon : B^A \times A \to B$ which evaluates a morphism $f \in B^A$ at an argument $x \in A$ to give a value $\epsilon \circ \langle f, x \rangle \in B$. This is always going to be present for the underlying functions if we’re starting from a set of functions $\mathsf{Hom}(A,B)$, but it needs to be an actual morphism in the category. Finally, we need an operation of “transposition”, taking a morphism $f : C \times A \to B$ to one $\widetilde{f} : C \to A^B$. We shall see that this in fact separates the previous two examples.

Definition 3.2.1. In a category $\mathcal{C}$ with binary products, an *exponential* (B^A, ϵ) of objects A and B is an object B^A together with a morphism $\epsilon : B^A \times A \to B$, called the *evaluation* morphism, such that for every $f : C \times A \to B$ there exists a *unique* morphism $\widetilde{f} : C \to B^A$, called the *transpose*[1] of f, for which the following diagram commutes.

$$\begin{tikzcd}
B^A & B^A \times A \arrow[r, "\epsilon"] & B \\
C \arrow[u, dotted, "\widetilde{f}"] & C \times A \arrow[u, "\widetilde{f} \times 1_A"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Commutativity of the diagram of course means that $f = \epsilon \circ (\widetilde{f} \times 1_A)$.

Definition 3.2.1 is called the *universal property of the exponential.* It is just the category-theoretic way of saying that a function $f : C \times A \to B$ of two variables can be viewed as a function $\widetilde{f} : C \to B^A$ of one variable that maps $z \in C$ to a function $\widetilde{f}z = f\langle z, -\rangle : A \to B$ that maps $x \in A$ to $f\langle z, x\rangle$. The relationship between f and $\widetilde{f}$ is then

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

That is all there is to it, really, except that variables and elements never need to be mentioned. The benefit of this is that the definition is applicable also in categories whose objects are not *sets* and whose morphisms are not *functions*—even though some of the basic examples are of that sort.

In Poset the exponential Q^P of posets P and Q is the set of all monotone maps $P \to Q$, ordered pointwise, as above. The evaluation map $\epsilon : Q^P \times P \to Q$ is just the usual evaluation of a function at an argument. The transpose of a monotone map $f : R \times P \to Q$ is the map $\widetilde{f} : R \to Q^P$, defined by, $(\widetilde{f}z)x = f\langle z, x\rangle$, i.e. the transposed *function.* We say that the category Pos *has all exponentials.*

Definition 3.2.2. Suppose $\mathcal{C}$ has all finite products. An object $A \in \mathcal{C}$ is *exponentiable* when the exponential B^A exists for every $B \in \mathcal{C}$. We say that $\mathcal{C}$ *has exponentials* if every object is exponentiable. A *cartesian closed category (ccc)* is a category that has all finite products and exponentials.

Example 3.2.3. Consider again the example of the set $\mathsf{Hom}(M, N)$ of homomorphisms between two monoids M, N, equipped with the pointwise monoid structure. To be a monoid homomorphism. the transpose $\widetilde{h} : 1 \to \mathsf{Hom}(M, N)$ of a homomorphism $h : 1 \times M \to N$ would have to take the unit element $u \in 1$ to the unit homomorphism $u : M \to N$, which is the constant function at the unit $u \in N$. Since $1 \times M \cong M$, that would mean that *all* homomorphisms $h : M \to N$ would have the same transpose $\widetilde{h} = u : 1 \to \mathsf{Hom}(M, N)$. So Mon cannot be cartesian closed. The same argument works in the category Group, and in many related ones. (But see **??** below on one way of embedding Group into a CCC.)

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

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

Two characterizations of CCCs

Proposition 3.2.5. *In a category $\mathcal{C}$ with binary products an object A is exponentiable if, and only if, the functor*

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

has a right adjoint

$$-^A : \mathcal{C} \to \mathcal{C} \; .$$

Proof. If such a right adjoint exists then the exponential of A and B is (B^A, ϵ_B), where $\epsilon : -^A \times A \Longrightarrow 1_{\mathcal{C}}$ is the counit of the adjunction. The universal property of the exponential is precisely the universal property of the counit ϵ.

Conversely, suppose for every B there is an exponential (B^A, ϵ_B). As the object part of the right adjoint we then take B^A. For the morphism part, given $g : B \to C$, we can define $g^A : B^A \to C^A$ to be the transpose of $g \circ \epsilon_B$,

$$g^A = (g \circ \epsilon_B)^{\sim}$$

as indicated below.

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

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

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

The universal property of the counit ϵ is precisely the universal property of the exponential (B^A, ϵ_B) □

Note that because exponentials can be expressed as right adjoints to binary products, they are determined uniquely up to isomorphism. Moreover, the definition of a cartesian closed category can then be phrased entirely in terms of adjoint functors: we just need to require the existence of the terminal object, binary products, and exponentials.

Proposition 3.2.6. *A category $\mathcal{C}$ is cartesian closed if, and only if, the following functors have* right *adjoints:*

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

Here $!_{\mathcal{C}}$ is the unique functor from $\mathcal{C}$ to the terminal category 1 and Δ is the diagonal functor $\Delta A = \langle A, A \rangle$, and the right adjoint of $- \times A$ is exponentiation by A.

$\square$

The significance of the adjoint formulation is that it implies the possibility of a purely *equational* specification (adjoint structure on a category is "equational" in a sense that can be made precise; see [**?**]). We can therefore give an explicit, equational formulation of cartesian closed categories.

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

1. *An object $1 \in \mathcal{C}$ and a morphism $!_A : A \to 1$ for every $A \in \mathcal{C}$.*
2. *An object $A \times B$ for all $A, B \in \mathcal{C}$ together with morphisms $\pi_0 : A \times B \to A$ and $\pi_1 : A \times B \to B$, and for every pair of morphisms $f : C \to A$, $g : C \to B$ a morphism $\langle f, g\rangle : C \to A \times B$.*
3. *An object B^A for all $A, B \in \mathcal{C}$ together with a morphism $\epsilon : B^A \times A \to B$, and a morphism $\widetilde{f} : C \to B^A$ for every morphism $f : C \times A \to B$.*

These new objects and morphisms are required to satisfy the following equations:

1. *For every $f : A \to 1$,*
$$f = \,!_A \; .$$
2. *For all $f : C \to A$, $g : C \to B$, $h : C \to A \times B$,*
$$\pi_0 \circ \langle f, g\rangle = f \; , \qquad \pi_1 \circ \langle f, g\rangle = g \; , \qquad \langle \pi_0 \circ h, \pi_1 \circ h\rangle = h \; .$$
3. *For all $f : C \times A \to B$, $g : C \to B^A$,*
$$\epsilon \circ (\widetilde{f} \times 1_A) = f \; , \qquad (\epsilon \circ (g \times 1_A))^{\sim} = g \; .$$
where for $e : E \to E'$ and $f : F \to F'$ we define
$$e \times f := \langle e\pi_0, f\pi_1\rangle : E \times F \to E' \times F'.$$

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

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

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

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

Suppose $h : C \to A \times B$ is another morphism such that $f = \pi_0 \circ h$ and $g = \pi_1 \circ h$. Then by the third equation for products we get

$$h = \langle \pi_0 \circ h, \pi_1 \circ h \rangle = \langle f, g \rangle \ ,$$

and so $\langle f, g \rangle$ is unique.

Exercise 3.2.8. Use the equational characterization of CCCs, Proposition 3.2.7, to show that the category Pos of posets and monotone functions *is* cartesian closed, as claimed. Also verify that that Mon is not. Which parts of the definition fail in Mon?

3.3 Positive propositional calculus

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

Example 3.3.1. Consider the *positive propositional calculus* PPC with conjunction and implication, as in Section **??**. Recall that PPC is the set of all propositional formulas ϕ constructed from propositional variables $p_1, p_2, \ldots$, a constant $\top$ for truth, and binary connectives for conjunction $\phi \wedge \psi$, and implication $\phi \Rightarrow \psi$.

As a category, PPC is a preorder under the relation $\phi \vdash \psi$ of logical entailment, determined for instance by the natural deduction system **??** of section **??**. As usual, it will be convenient to pass to the poset reflection of the preorder, which we shall denote by

$$\mathcal{C}_{\mathsf{PPC}}$$

by identifying ϕ and ψ when $\phi \dashv\vdash \psi$. (This is just the usual *Lindenbaum-Tarski* algebra of the system of propositional logic, as in Section **??**.)

The conjunction $\phi \wedge \psi$ is a greatest lower bound of ϕ and ψ in $\mathcal{C}_{\mathsf{PPC}}$, because we have $\phi \wedge \psi \vdash \phi$ and $\phi \wedge \psi \vdash \psi$ and for all ϑ, if $\vartheta \vdash \phi$ and $\vartheta \vdash \psi$ then $\vartheta \vdash \phi \wedge \psi$. Since binary products in a poset are the same thing as greatest lower bounds, we see that $\mathcal{C}_{\mathsf{PPC}}$ has all binary products; and of course $\top$ is a terminal object.

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

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

Therefore $\phi \Rightarrow \psi$ is an exponential in $\mathcal{C}_{\mathsf{PPC}}$. The counit of the adjunction (the "evaluation" arrow) is the entailment

$$(\phi \Rightarrow \psi) \wedge \phi \vdash \psi \ ,$$

i.e. the familiar logical rule of *modus ponens.*

We have now shown:

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

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

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

$$k \Vdash p$$

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

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

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

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

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

$$K \Vdash \phi$$

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

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

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

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

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

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

preserves any CCC structure that exists in P.

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

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

Then for any downset Q we have

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

But that means that

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

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

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

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

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

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

where the last line is by (3.5). (Note that in line (3) we assumed that $d \wedge p$ exists for all $d \in D$; this can be avoided by a slightly more complicated argument.) □

Proof. (of Theorem 3.3.4) The proof follows a now-familiar pattern, which we only sketch:

1. The syntactic category $\mathcal{C}_{\mathsf{PPC}}$ is a CCC, with $\top = 1$, $\phi \times \psi = \phi \wedge \psi$, and $\psi^{\phi} = \phi \Rightarrow \psi$. In fact, it is the free cartesian closed poset on the generating set $\mathsf{Var} = \{p_1, p_2, \dots\}$ of propositional variables.

2. A (Kripke) model $(K, \Vdash)$ is the same thing as a CCC functor $\mathcal{C}_{\mathsf{PPC}} \to {\downarrow}K$, which by Step 1 is just an arbitrary map $\mathsf{Var} \to {\downarrow}K$, as in (3.3). To see this, observe that we have a bijective correspondence between CCC functors $[\![-]\!]$ and Kripke relations $\Vdash$; indeed, by the exponential adjunction in the cartesian closed category Pos, there is a natural bijection,
$$\frac{[\![-]\!] : \mathcal{C}_{\mathsf{PPC}} \longrightarrow {\downarrow}K \cong \mathbb{2}^{K^{\mathsf{op}}}}{\Vdash : K^{\mathsf{op}} \times \mathcal{C}_{\mathsf{PPC}} \longrightarrow \mathbb{2}}$$
where we use the poset $\mathbb{2}$ to classify downsets in a poset P (via upsets in P^{op}),
$${\downarrow}P \cong \mathbb{2}^{P^{\mathsf{op}}} \cong \mathsf{Pos}(P^{\mathsf{op}}, \mathbb{2})\,,$$
by taking the 1-kernel $f^{-1}(1) \subseteq P$ of a monotone map $f : P^{\mathsf{op}} \to \mathbb{2}$. (The contravariance will be convenient in Step 3). Note that the monotonicity of $\Vdash$ yields the conditions
$$p \leq q\,,\ q \Vdash \phi \implies p \Vdash \phi$$

and

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

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

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

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

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

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

□

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

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

3.4 Heyting algebras

We now extend the positive propositional calculus to the full intuitionistic propositional calculus. This involves adding the finite coproducts 0 and $p \vee q$ to notion of a cartesian closed poset, to arrive at the general notion of a Heyting algebra. Heyting algebras are to intuitionistic logic as Boolean algebras are to classical logic: each is an algebraic description of the corresponding logical calculus. We shall review both the algebraic and the logical points of view; as we shall see, many aspects of the theory of Boolean algebras carry over to Heyting algebras. For instance, in order to prove the Kripke completeness of the full system of intuitionistic propositional calculus, we will need an alternative to Lemma 3.3.5, because the Yoneda embedding does not in general preserve coproducts. For that we will again use a version of the Stone representation theorem, this time in a generalized form due to Joyal.

Distributive lattices

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

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

A *lattice homomorphism* is a function $f : L \to K$ between lattices which preserves finite limits and colimits, i.e., $f0 = 0$, $f1 = 1$, $f(x \wedge y) = fx \wedge fy$, and $f(x \vee y) = fx \vee fy$. The category of lattices and lattice homomorphisms is denoted by Lat.

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

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

The partial order on L is then determined by

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

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

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

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

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

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

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

A *Heyting algebra homomorphism* is a lattice homomorphism $f : K \to H$ between Heyting algebras that preserves implication, i.e., $f(x \Rightarrow y) = (fx \Rightarrow fy)$. The category of Heyting algebras and their homomorphisms is denoted by Heyt.

Heyting algebras can be axiomatized equationally as a set H with two distinguished elements 0 and 1 and three binary operations $\wedge$, $\vee$ and $\Rightarrow$. The equations for a Heyting algebra are the ones listed in (3.6), as well as the following ones for $\Rightarrow$.

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

For a proof, see [Joh82, I.1], where one can also find a proof that every Heyting algebra is distributive (exercise!).

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

Intuitionistic propositional calculus

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

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

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

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

3. Connectives: if $\phi, \psi \in HS$ then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \Rightarrow \psi) \in HS$.

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

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

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

The inclusion $\eta_S : S \to U(HS)$ of generators into the underlying set of the free Heyting algebra HS is then the component at S of a natural transformation $\eta : 1_{\mathsf{Set}} \Longrightarrow U \circ H$, which is of course the unit of the adjunction $H \dashv U$. To see this, consider a Heyting algebra K and an arbitrary function $f : S \to UK$. Then the Heyting algebra homomorphism $\overline{f} : HS \to K$ defined by

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

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

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

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

We may now define the *intuitionistic propositional calculus* IPC to be the free Heyting algebra IPC on countably many generators p_0, p_1, ..., called *atomic propositions* or *propositional variables*. This is a somewhat unorthodox definition from a logical point of view—normally we would start from a *calculus* consisting of a formal language, judgements, and rules of inference—but of course, by now, we realize that the two approaches are essentially equivalent.

Having said that, let us also describe IPC in the conventional way. The formulas of IPC are built inductively from propositional variables p_0, p_1, ..., constants falsehood $\bot$ and truth $\top$, and binary operations conjunction $\wedge$, disjunction $\vee$ and implication $\Rightarrow$. The basic judgment of IPC is *logical entailment*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Classical propositional calculus

Another look:

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

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

We say that y is the *complement* of x.

In a distributive lattice, the complement of x is unique if it exists. Indeed, if both y and z are complements of x then

$$y \wedge z = (y \wedge z) \vee 0 = (y \wedge z) \vee (y \wedge x) = y \wedge (z \vee x) = y \wedge 1 = y ,$$

hence $y \leq z$. A symmetric argument shows that $z \leq y$, therefore $y = z$. The complement of x, if it exists, is denoted by $\neg x$.

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

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

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

Exercise 3.4.6. Prove that every Boolean algebra is a Heyting algebra. Hint: how is implication encoded in terms of negation and disjunction in classical logic?

In a Heyting algebra not every element is complemented. However, we can still define a *pseudo complement* or *negation* operation $\neg$ by

$$\neg x = (x \Rightarrow 0) ,$$

Then $\neg x$ is the largest element for which $x \wedge \neg x = 0$. While in a Boolean algebra $\neg\neg x = x$, in a Heyting algebra we only have $\neg\neg x \leq x$ in general. An element x of a Heyting algebra for which $\neg\neg x = x$ is called a *regular* element.

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

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

Exercise 3.4.8. Prove that the topology $\mathcal{O}X$ of any topological space X is a Heyting algebra. Describe in topological language the implication $U \Rightarrow V$, the negation $\neg U$, and the regular elements $U = \neg\neg U$ in $\mathcal{O}X$.

Exercise 3.4.9. Show that for a Heyting algebra H, the regular elements of H form a Boolean algebra $H_{\neg\neg} = \{x \in H \mid x = \neg\neg x\}$. Here $H_{\neg\neg}$ is viewed as a subposet of H. Hint: negation $\neg'$, conjunction $\wedge'$, and disjunction $\vee'$ in $H_{\neg\neg}$ are expressed as follows in terms of negation, conjunction and disjunction in H, for $x, y \in H_{\neg\neg}$:

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

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

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

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

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

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

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

Exercise 3.4.11. Show that in a Heyting algebra H, one has $\neg\neg x = x$ for all $x \in H$ if, and only if, $y \vee \neg y = 1$ for all $y \in H$. Hint: half of the equivalence is easy. For the other half, observe that the assumption $\neg\neg x = x$ means that negation is an order-reversing bijection $H \to H$. It therefore transforms joins into meets and vice versa, and so the *De Morgan laws* hold:

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

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

Kripke semantics for IPC

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

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

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

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

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

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

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

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

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

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

would not preserve the coproducts $\bot$ and $\phi \vee \psi$. Indeed, $\downarrow(\bot) \neq \emptyset$, because it contains $\bot$ itself! And in general $\downarrow(\phi \vee \psi) \neq \downarrow(\phi) \cup \downarrow(\psi)$, because the righthand side need not contain, e.g., $\phi \vee \psi$.

Instead, we will generalize the Stone Representation theorem **??** from Boolean algebras to Heyting algebras, using a theorem due to Joyal (cf. [MR95, MH92]). First, recall that the Stone representation provided, for any Boolean algebra $\mathcal{B}$, an injective Boolean homomorphism into a powerset,

$$\mathcal{B} \rightarrowtail \mathcal{P}X\,.$$

For X we took the set of prime filters $\mathsf{Bool}(\mathcal{B}, \mathbb{2})$, and the map $h : \mathcal{B} \rightarrowtail \mathcal{P}\mathsf{Bool}(\mathcal{B}, \mathbb{2})$ was given by $h(b) = \{F \mid b \in F\}$. Transposing $\mathcal{P}\mathsf{Bool}(\mathcal{B}, \mathbb{2}) \cong \mathbb{2}^{\mathsf{Bool}(\mathcal{B}, \mathbb{2})}$ in the cartesian closed category Pos, we arrive at the (monotone) evaluation map

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

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

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

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

The transpose of the (monotone) evaluation map,

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

will then be the (monotone) map

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

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

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

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

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

which generalizes the Stone representation from Boolean to Heyting algebras.

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

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

into a Heyting algebra of upsets in a poset J.

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

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

which, as before, is injective, by the Prime Ideal Theorem (see Lemma **??**). We will use it in the following form due to Birkhoff.

Lemma 3.4.14 (Birkhoff's Prime Ideal Theorem)**.** *Let D be a distributive lattice, $I \subseteq D$ an ideal, and $x \in D$ with $x \notin I$. There is a prime ideal $I \subseteq P \subset D$ with $x \notin P$.*

Proof. As in the proof of Lemma **??**, it suffice to prove it for the case $I = (0)$. This time, we use Zorn's Lemma: a poset in which every chain has an upper bound has maximal elements. Consider the poset $\mathcal{I}\backslash x$ of "ideals I without x", $x \notin I$, ordered by inclusion.

The union of any chain $I_0 \subseteq I_1 \subseteq \ldots$ in $\mathcal{I}\backslash x$ is clearly also in $\mathcal{I}\backslash x$, so we have (at least one) maximal element $M \in \mathcal{I}\backslash x$. We claim that $M \subseteq D$ is prime. To that end, take $a, b \in D$ with $a \wedge b \in M$. If $a, b \notin M$, let $M[a] = \{n \leq m \vee a \mid m \in M\}$, the ideal join of M and $\downarrow(a)$, and similarly for $M[b]$. Since M is maximal without x, we therefore have $x \in M[a]$ and $x \in M[b]$. Thus let $x \leq m \vee a$ and $x \leq m' \vee b$ for some $m, m' \in M$. Then $x \vee m' \leq m \vee m' \vee a$ and $x \vee m \leq m \vee m' \vee b$, so taking meets on both sides gives

$$(x \vee m') \wedge (x \vee m) \leq (m \vee m' \vee a) \wedge (m \vee m' \vee b) = (m \vee m') \vee (a \wedge b).$$

Since the righthand side is in the ideal M, so is the left. But then $x \leq x \vee (m \wedge m')$ is also in M, contrary to our assumption that $M \in \mathcal{I}\backslash x$. □

Proof of Theorem 3.4.13. As in (3.17), let $J = \mathsf{DLat}(\mathcal{H}, \mathbb{2})$ be the poset of prime filters in $\mathcal{H}$, and consider the "evaluation" map (3.17),

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

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

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

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

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

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

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

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

Now if $p \Rightarrow q \in F$, then for all (prime) filters $G \supseteq F$, also $p \Rightarrow q \in G$, and so $p \in G$ implies $q \in G$, since $(p \Rightarrow q) \wedge p \leq q$.

Conversely, suppose $p \Rightarrow q \notin F$, and we seek a prime filter $G \supseteq F$ with $p \in G$ but $q \notin G$. Consider the filter

$$F[p] = \{x \wedge p \leq h \in \mathcal{H} \mid x \in F\}\,,$$

which is the join of F and $\uparrow(p)$ in the poset of filters. If $q \in F[p]$, then $x \wedge p \leq q$ for some $x \in F$, whence $x \leq p \Rightarrow q$, and so $p \Rightarrow q \in F$, contrary to assumption; thus $q \notin F[p]$. By the Prime Ideal Theorem again (applied to the distributive lattice $\mathcal{H}^{\mathsf{op}}$) there is a prime filter $G \supseteq F[p]$ with $q \notin G$. □

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

3.5 Frames and spaces

A poset $(P, \leq)$, viewed as a category, is *cocomplete* when it has suprema (least upper bounds) of arbitrary subsets. This is so because coequalizers in a poset always exist, and coproducts are precisely least upper bounds. Recall that the supremum of $S \subseteq P$ is an element $\bigvee S \in P$ such that, for all $y \in S$,

$$\bigvee S \leq y \iff \forall\, x : S \,.\, x \leq y \,.$$

In particular, $\bigvee \emptyset$ is the least element of P and $\bigvee P$ is the greatest element of P. Similarly, a poset is *complete* when it has infima (greatest lower bounds) of arbitrary subsets; the infimum of $S \subseteq P$ is an element $\bigwedge S \in P$ such that, for all $y \in S$,

$$y \leq \bigwedge S \iff \forall\, x : S \,.\, y \leq x \,.$$

Proposition 3.5.1. *A poset is complete if, and only if, it is cocomplete.*

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

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

□

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

Suppose P is a complete poset. When is it cartesian closed? Being a complete poset, it has the terminal object, namely the greatest element $1 \in P$, and it has binary products which are binary infima. If P is cartesian closed then for all $x, y \in P$ there exists an exponential $(x \Rightarrow y) \in P$, which satisfies, for all $z \in P$,

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

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

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

as follows:

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

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

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

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

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

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

and that, for $z \in P$,

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

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

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

Complete cartesian closed posets are called *frames*.

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

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

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

Warning: a frame morphism need not preserve exponentials!

Example 3.5.3. Given a poset P, the downsets $\downarrow P$ form a complete lattice under the inclusion order $S \subseteq T$, and with the set theoretic operations $\bigcup$ and $\bigcap$ as $\bigvee$ and $\bigwedge$. Since $\downarrow P$ is already known to be a Heyting algebra (Example 3.4.2), it is therefore also a frame. (Alternately, we can show that it is a frame by noting that the operations $\bigcup$ and $\bigcap$ satisfy the infinite distributive law, and then infer that it is a Heyting algebra.)

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

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

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

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

Example 3.5.4. The topology $\mathcal{O}X$ of a topological space X, ordered by inclusion, is a frame because finite intersections and arbitrary unions of open sets are open. The distributive law holds because intersections distribute over unions. If $f : X \to Y$ is a continuous map between topological spaces, the inverse image map $f^* : \mathcal{O}Y \to \mathcal{O}X$ is a frame homomorphism. Thus, there is a functor

$$\mathcal{O} : \mathsf{Top} \to \mathsf{Frame}^{\mathsf{op}}$$

which maps a space X to its topology $\mathcal{O}X$ and a continuous map $f : X \to Y$ to the inverse image map $f^* : \mathcal{O}Y \to \mathcal{O}X$.

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

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

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

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

Exercise* 3.5.6. This exercise is meant for students with some background in topology. For a topological space X and a point $x \in X$, let $N(x)$ be the neighborhood filter of x,

$$N(x) = \big\{ U \in \mathcal{O}X \bigm| x \in U \big\} \ .$$

Recall that a T_0-*space* is a topological space X in which points are determined by their neighborhood filters,

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

Let Top_0 be the full subcategory of Top on T_0-spaces. The functor $\mathcal{O} : \mathsf{Top} \to \mathsf{Loc}$ restricts to a functor $\mathcal{O} : \mathsf{Top}_0 \to \mathsf{Loc}$. Prove that $\mathcal{O} : \mathsf{Top}_0 \to \mathsf{Loc}$ is a faithful functor. Is it full?

Topological semantics for IPC

It should now be clear how to interpret IPC into a topological space X: each formula ϕ is assigned to an open set $[\![\phi]\!] \in \mathcal{O}X$ in such a way that $[\![-]\!]$ is a homomorphism of Heyting algebras.

Definition 3.5.7. A *topological model* of IPC is a space X and an interpretation of formulas,

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

satisfying the conditions:

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

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

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

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

Corollary 3.5.8. *A formula ϕ is provable in* IPC *if, and only if, it holds in every topological interpretation $[\![-]\!]$ into a space X, briefly:*

$$\mathsf{IPC} \vdash \phi \qquad \textit{iff} \qquad [\![\phi]\!] = X \textit{ for all spaces } X \,.$$

Proof. Put the Alexandrov topology on the upsets of prime ideals in the Heyting algebra IPC □

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

3.6 Proper CCCs

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

Example 3.6.1. The first example is the category Set. We already know that the terminal object is a singleton set and that binary products are cartesian products. The exponential of X and Y in Set is just the set of all functions from X to Y,

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

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

Example 3.6.2. The category Cat of all small categories is cartesian closed. The exponential of small categories $\mathcal{C}$ and $\mathcal{D}$ is the category $\mathcal{D}^{\mathcal{C}}$ of functors, with natural transformations as arrows (see ??). Note that if $\mathcal{D}$ is a groupoid (all arrows are isos), then so is $\mathcal{D}^{\mathcal{C}}$. It follows that the category of groupoids is full (even as a 2-category) in Cat. Since limits of groupoids in Cat are also groupoids, the inclusion of the full subcategory $\mathsf{Grpd} \hookrightarrow \mathsf{Cat}$ preserves limits, too, and is therefore a full inclusion of CCCs.

Example 3.6.3. The same reasoning as in the previous example shows that the full subcategory $\mathsf{Pos} \hookrightarrow \mathsf{Cat}$ of all small posets and monotone maps is also cartesian closed. It is worth noting that, unlike the previous cases, the (limit preserving) forgetful functor $U : \mathsf{Poset} \to \mathsf{Set}$ does *not* preserve exponentials; in general $U(Q^P) \subseteq (UQ)^{UP}$ is a *proper* subset.

Exercise 3.6.4. There is a full and faithful functor $I : \mathsf{Set} \to \mathsf{Poset}$ that preserves finite limits as well as exponentials. How is this related to the example $\mathsf{Grpd} \hookrightarrow \mathsf{Cat}$?

The foregoing examples are instances of the following general situation.

Proposition 3.6.5. *Let $\mathcal{E}$ be a CCC and $i : \mathcal{S} \hookrightarrow \mathcal{E}$ a full subcategory with finite products and a left adjoint reflection $L : \mathcal{E} \to \mathcal{S}$ preserving finite products. Suppose moreover that for any two objects A, B in $\mathcal{S}$, the exponential iB^{iA} is again in $\mathcal{S}$. Then $\mathcal{S}$ has all exponentials, and these are preserved by i.*

Proof. By assumption, we have $L \dashv i$ with isomorphic counit $LiS \cong S$ for all $S \in \mathcal{S}$. Let us identify $\mathcal{S}$ with the subcategory of $\mathcal{E}$ that is its image under $i : \mathcal{S} \hookrightarrow \mathcal{E}$. The assumption that B^A is again in $\mathcal{S}$ for all $A, B \in \mathcal{S}$, along with the fullness of $\mathcal{S}$ in $\mathcal{E}$, gives the exponentials, and the closure of $\mathcal{S}$ under finite products in $\mathcal{E}$ ensures that the required transposes will also be in $\mathcal{S}$.

Alternately, for any $A, B \in \mathcal{S}$ set $B^A = L(iB^{iA})$. Then for any $C \in \mathcal{S}$, we have natural isos:

$$\begin{aligned}
\mathcal{S}(C \times A, B) &\cong \mathcal{E}(i(C \times A), iB) \\
&\cong \mathcal{E}(iC \times iA, iB) \\
&\cong \mathcal{E}\big(iC, iB^{iA}\big) \\
&\cong \mathcal{E}\big(iC, iL(iB^{iA})\big) \\
&\cong \mathcal{S}\big(C, L(iB^{iA})\big) \\
&\cong \mathcal{S}\big(C, B^A\big)
\end{aligned}$$

where in the fifth line we used the assumption that iB^{iA} is again in $\mathcal{S}$, in the form $iB^{iA} \cong iE$ for some $E \in \mathcal{S}$, which is then necessarily $L(iB^{iA}) = LiE \cong E$. □

A related general situation that covers some (but not all) of the above examples is this:

Proposition 3.6.6. *Let $\mathcal{E}$ be a CCC and $i : \mathcal{S} \hookrightarrow \mathcal{E}$ a full subcategory with finite products and a right adjoint reflection $R : \mathcal{E} \to \mathcal{S}$. If i preserves finite products, then $\mathcal{S}$ also has all exponentials, and these are computed first in $\mathcal{E}$, and then reflected by R into $\mathcal{S}$.*

Proof. For any $A, B \in \mathcal{S}$ set $B^A = R(iB^{iA})$ as described. Now for any $C \in \mathcal{S}$, we have natural isos:

$$\begin{aligned} \mathcal{S}(C \times A, B) &\cong \mathcal{E}(i(C \times A), iB) \\ &\cong \mathcal{E}(iC \times iA, iB) \\ &\cong \mathcal{E}\big(iC, iB^{iA}\big) \\ &\cong \mathcal{S}\big(C, R(iB^{iA})\big) \\ &\cong \mathcal{S}\big(C, B^{A}\big) . \end{aligned}$$

□

An example of the foregoing is the inclusion of the opens into the powerset of points of a space X,

$$\mathcal{O}X \hookrightarrow \mathcal{P}X$$

This frame homomorphism is associated to the map $|X| \to X$ of locales (or in this case, spaces) from the discrete space on the set of points of X.

Exercise 3.6.7. Which of the examples follows from which proposition?

Example 3.6.8. A presheaf category $\widehat{\mathbb{C}}$ is cartesian closed, provided the index category $\mathbb{C}$ is small. To see what the exponential of presheaves P and Q ought to be, we use the Yoneda Lemma. If Q^P exists, then by Yoneda Lemma and the adjunction $(- \times P) \dashv (-^P)$, we have for all $A \in \mathbb{C}$,

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

Because $\mathcal{C}$ is small $\mathsf{Nat}(\mathsf{y}A \times P, Q)$ is a set, so we can *define* Q^P to be the presheaf

$$Q^P = \mathsf{Nat}(\mathsf{y}{-} \times P, Q) .$$

The evaluation morphism $E : Q^P \times P \Longrightarrow Q$ is the natural transformation whose component at A is

$$\begin{aligned} &E_A : \mathsf{Nat}(\mathsf{y}A \times P, Q) \times PA \to QA , \\ &E_A : \langle \eta, x \rangle \mapsto \eta_A \langle 1_A, x \rangle . \end{aligned}$$

The transpose of a natural transformation $\phi : R \times P \Longrightarrow Q$ is the natural transformation $\widetilde{\phi} : R \Longrightarrow Q^P$ whose component at A is the function that maps $z \in RA$ to the natural transformation $\widetilde{\phi}_A z : \mathsf{y}A \times P \Longrightarrow Q$, whose component at $B \in \mathcal{C}$ is

$$\begin{aligned} &(\widetilde{\phi}_A z)_B : \mathcal{C}(B, A) \times PB \to QB , \\ &(\widetilde{\phi}_A z)_B : \langle f, y \rangle \mapsto \phi_B \langle (Rf)z, y \rangle . \end{aligned}$$

Exercise 3.6.9. Verify that the above definition of Q^P really gives an exponential of presheaves P and Q.

It follows immediately that the category of graphs Graph is cartesian closed because it is the presheaf category $\mathsf{Set}^{\cdot\rightrightarrows\cdot}$. The same is of course true for the "category of functions", i.e. the arrow category $\mathsf{Set}^{\rightarrow}$, as well as the category of simplicial sets $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ from topology.

Exercise 3.6.10. This exercise is for students with some background in linear algebra. Let Vec be the category of real vector spaces and linear maps between them. Given vector spaces X and Y, the linear maps $\mathcal{L}(X, Y)$ between them form a vector space. So define $\mathcal{L}(X, -) : \mathsf{Vec} \to \mathsf{Vec}$ to be the functor which maps a vector space Y to the vector space $\mathcal{L}(X, Y)$, and it maps a linear map $f : Y \to Z$ to the linear map $\mathcal{L}(X, f) : \mathcal{L}(X, Y) \to \mathcal{L}(X, Z)$ defined by $h \mapsto f \circ h$. Show that $\mathcal{L}(X, -)$ has a left adjoint $- \otimes X$, but also show that this adjoint is *not* the binary product in Vec.

A few other instructive examples that can be explored by the interested reader are the following.

- Etale spaces over a base space X. This category can be described as consisting of *local homeomorphisms* $f : Y \to X$ and commutative triangles over X between such maps. It is also equivalent to the category $\mathsf{Sh}(X)$ of *sheaves* on X. See [**?**, ch.n].
- Various subcategories of topological spaces (sequential spaces, compactly-generated spaces). Cf. [**?**].
- Dana Scott's category Equ of equilogical spaces [**?**].

3.7 Simply typed λ-calculus

The λ-calculus is the abstract theory of functions, just like group theory is the abstract theory of symmetries. There are two basic operations that can be performed with functions. The first one is the *application* of a function to an argument: if f is a function and a is an argument, then fa is the application of f to a, also called the *value* of f at a. The second operation is *abstraction*: if x is a variable and t is an expression in which x may appear, then there is a function f defined by the equation

$$fx = t\ .$$

Here we gave the name f to the newly formed function. But we could have expressed the same function without giving it a name; this is usually written as

$$x \mapsto t\ ,$$

and it means "x is mapped to t". In λ-calculus we use a different notation, which is more convenient when abstractions are nested:

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

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

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

Remark 3.7.1. It may seem strange that in specifying the abstraction of a function, we switched from talking about objects (functions, arguments, values) to talking about *expressions*: variables, names, equations. This "syntactic" point of view seems to have been part of the notion of a function since its beginnings, in the theory of algebraic equations. It is the reason that the λ-calculus is part of *logic*, unlike the theory of cartesian closed categories, which remains thoroughly semantical (and "variable-free"). The relation between the two different points of view will occupy the remainder of this chapter.

Remark 3.7.2. There are two kinds of λ-calculus, the *typed* and the *untyped* one. In the untyped version there are no restrictions on how application is formed, so that an expression such as

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

is valid, whatever it may mean. In typed λ-calculus every expression has a *type*, and there are rules for forming valid expressions and types. For example, we can only form an application f, a when a has a type A and f has a type $A \to B$, which indicates a function taking arguments of type A and giving results of type B. The judgment that expression t has a type A is written as

$$t : A\ .$$

To computer scientists the idea of expressions having types is familiar from programming languages, whereas mathematicians can think of types as sets and read $t : A$ as $t \in A$. We will concentrate on the typed λ-calculus.

We now give a precise definition of what constitutes a *simply-typed λ-calculus*. First, we are given a set of *simple types*, which are generated from *basic types* by formation of products and function types:

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

Function types associate to the right:

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

We assume there is a countable set of *variables* x, y, u, ... We are also given a set of *basic constants*. The set of *terms* is generated from variables and basic constants by the following grammar:

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

In words, this means:

1. a variable is a term,

2. each basic constant is a term,
3. the constant $*$ is a term, called the *unit*,
4. if u and t are terms then $\langle u, t\rangle$ is a term, called a *pair*,
5. if t is a term then $\mathtt{fst}\,t$ and $\mathtt{snd}\,t$ are terms,
6. if u and t are terms then $u\,t$ is a term, called an *application*
7. if x is a variable, A is a type, and t is a term, then $\lambda x : A\,.\,t$ is a term, called a λ-*abstraction*.

The variable x is *bound* in $\lambda x : A\,.\,t$. Application associates to the left, thus $s\,t\,u = (s\,t)\,u$. The *free variables* $\mathsf{FV}(t)$ of a term t are computed as follows:

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

If $x_1, \ldots, x_n$ are *distinct* variables and $A_1, \ldots, A_n$ are types then the sequence

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

is a *typing context*, or just *context*. The empty sequence is sometimes denoted by a dot $\cdot$, and it is a valid context. Context are denoted by capital Greek letters Γ, Δ, …

A *typing judgment* is a judgment of the form

$$\Gamma \mid t : A$$

where Γ is a context, t is a term, and A is a type. In addition the free variables of t must occur in Γ, but Γ may contain other variables as well. We read the above judgment as "in context Γ the term t has type A". Next we describe the rules for deriving typing judgments.

Each basic constant c_i has a uniquely determined type C_i,

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

The type of a variable is determined by the context:

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

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,

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

the judgment that they are equal is written as

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

Note that u and t necessarily have the same type; it does *not* make sense to compare terms of different types. We have the following rules for equations:

1. Equality is an equivalence relation:

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

2. The weakening rule:

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

3. Unit type:

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

4. Equations for product types:

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

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

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

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

5. Equations for function types:

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

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

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

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

This completes the description of a simply-typed λ-calculus.

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

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

We summarize the preceding definitions.

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

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

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

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

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 3.7.5. In general, any algebraic theory $\mathbb{A}$ determines a λ-theory $\mathbb{A}_\lambda$. There is one basic type $\mathtt{A}$ and for each operation f of arity k there is a basic constant $\mathtt{f} : \mathtt{A}^k \to \mathtt{A}$, where $\mathtt{A}^k$ is the k-fold product $\mathtt{A} \times \cdots \times \mathtt{A}$. It is understood that $\mathtt{A}^0 = \mathtt{1}$. The terms of $\mathbb{A}$ are translated to the terms of the corresponding λ-theory in a straightforward manner. For every axiom $t = u$ of $\mathbb{A}$ the corresponding axiom in the λ-theory is

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

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

Example 3.7.6. The theory of a directed graph is a simply-typed theory with two basic types, `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 3.7.7. An example of a λ-theory is readily found in the theory of programming languages. The mini-programming language PCF is a simply-typed λ-calculus with a basic type `nat` for natural numbers, and a basic type `bool` of Boolean values,

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

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

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

The equational axioms of PCF are:

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

Example 3.7.8. 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

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

which says that `s` is a section and `r` is a retraction, so that the function type $\mathtt{D} \to \mathtt{D}$ is a subspace (even a retract) of `D`. A type with this property is said to be *reflexive*. We may additionally stipulate the axiom

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

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

Untyped λ-calculus

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

$$t ::= v \mid t_! \, t_2 \mid \lambda x.\, t \; .$$

In words, a variable is a term, an application $t\,t'$ is a term, for any terms t and t', and a λ-abstraction $\lambda x.\, t$ is a term, for any term t. Variable x is bound in $\lambda x.\, t$. A *context* is a list of distinct variables,

$$x_1, \ldots, x_n \; .$$

We say that a term t is valid in context Γ if the free variables of t are listed in Γ. The judgment that two terms u and t are equal is written as

$$\Gamma \mid u = t \; ,$$

where it is assumed that u and t are both valid in Γ. The context Γ is not really necessary but we include it because it is always good practice to list the free variables.

The rules of equality are as follows:

1. Equality is an equivalence relation:

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

2. The weakening rule:

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

3. Equations for application and λ-abstraction:

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

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

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

The untyped λ-calculus can be translated into the theory of a reflexive type from Example 3.7.8. An untyped context Γ is translated to a typed context Γ^* by typing each variable in Γ with the reflexive type $\mathtt{D}$, i.e., a context $x_1, \ldots, x_k$ is translated to $x_1 : \mathtt{D}, \ldots, x_k : \mathtt{D}$. An untyped term t is translated to a typed term t^* as follows:

$$\begin{aligned} x^* &= x \qquad \text{if } x \text{ is a variable} \; , \\ (u\,t)^* &= (\mathtt{r}\,u^*)t^* \; , \\ (\lambda x.\, t)^* &= \mathtt{s}\,(\lambda x : \mathtt{D}\,.\, t^*) \; . \end{aligned}$$

For example, the term $\lambda x.\,(x\,x)$ translates to $\mathtt{s}\,(\lambda x : \mathtt{D}\,.\,((\mathtt{r}\,x)\,x))$. A judgment

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

is translated to the judgment

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

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

3.8 Interpretation of λ-calculus in CCCs

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

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

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

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

The interpretation is extended to all terms in context as follows. A context $\Gamma = x_1 : A_1, \cdots, x_n : A_n$ is interpreted as the object

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

and the empty context is interpreted as the terminal object 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

$$\begin{tikzcd} {[\![\Gamma]\!]} \arrow[r, "!_{[\![\Gamma]\!]}"] & 1 \arrow[r, "{[\![c]\!]}"] & {[\![A]\!]} \end{tikzcd}$$

3. The interpretation of projections and pairs is

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

4. The interpretation of application and λ-abstraction is

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

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

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

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

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

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

Proposition 3.8.1 (Soundness). *If $\mathbb{T}$ is a λ-theory and $[\![-]\!]$ a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every equation in context $\Gamma \mid t = u : A$ that is provable from the axioms of $\mathbb{T}$, we have*

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

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

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

Remark 3.8.2 (Inhabitation). There is another notion of provability for the λ-calculus, related to the Curry-Howard correspondence of section 3.1, relating it to propositional logic. If we regard types as "propositions" rather than structures, and terms as "proofs" rather than operations, then it is more natural to ask whether there even *is* a term $a : A$ of some type, than whether two terms of the same type are equal $s = t : A$. This only makes sense when A is considered in the empty context $\cdot \vdash A$, rather than $\Gamma \vdash A$ for non-empty Γ (consider the case where $\Gamma = x : A, \dots$). We say that a type A is *inhabited* (by a closed term) when there is some $\vdash a : A$, and regard an inhabited type A a *provable.* In this sense, there is another notion of soundness as follows.

Proposition 3.8.3 (Inhabitation soundness). *If $\mathbb{T}$ is a λ-theory and $[\![-]\!]$ a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every closed type A that is inhabited in $\mathbb{T}$ by a closed term, $\vdash a : A$, there is a corresponding point*

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

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

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

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

Example 3.8.4.

1. A model of an algebraic theory $\mathbb{A}$, extended to a λ-theory $\mathbb{A}_\lambda$ as in Example 3.7.5, taken in a CCC $\mathcal{C}$, is just a model of the algebraic theory $\mathbb{A}$ in the underlying finite product category $|\mathcal{C}|$ of $\mathcal{C}$. A difference, however, is that in defining the *category of models*
$$\mathsf{Mod}_\times(\mathbb{A}, |\mathcal{C}|)$$
we can take all homomorphism of models the algebraic theory as arrows, while the arrows in the category
$$\mathsf{Mod}_\lambda(\mathbb{A}_\lambda, \mathcal{C})$$
of λ-models are best taken to be isomorphisms, for which one has an obvious way to deal with the contravariance of the function type $[\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}$.

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

3. A model of a propositional theory $\mathbb{T}$, regarded as a λ-theory, in a CCC poset P is the same thing as before: an interpretation of the atomic propositions $p_1, p_2, ...$ of $\mathbb{T}$ as elements $[\![p_1]\!], [\![p_2]\!], ... \in P$, such that the axioms $\phi_1, \phi_2, ...$ of $\mathbb{T}$ are all sent to $1 \in P$ by the extension of $[\![-]\!]$ to all formulas, i.e. $[\![\phi_1]\!] = [\![\phi_2]\!] = ... = 1 \in P$.

3.9 Functorial semantics

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

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

- Morphisms $A \to B$ are terms in context
$$[x : A \mid t : B] \ ,$$
where two such terms $x : A \mid t : B$ and $x : A \mid u : B$ represent the same morphism when $\mathbb{T}$ proves $x : A \mid t = u : B$.

- Composition of the terms
$$[x : A \mid t : B] : A \longrightarrow B \qquad \text{and} \qquad [y : B \mid u : C] : B \longrightarrow C$$
is the term obtained by substituting t for y in u:
$$[x : A \mid u[t/y] : C] : A \longrightarrow C \ .$$

- The identity morphism on A is the term $[x : A \mid x : A]$.

Proposition 3.9.1. *The syntactic category $\mathcal{C}_{\mathbb{T}}$ built from a λ-theory is cartesian closed.*

Proof. We omit the equivalence classes brackets $[x : A \mid t : B]$ and treat equivalent terms as equal.

- The terminal object is the unit type $\mathtt{1}$. For any type A the unique morphism $!_A : A \to \mathtt{1}$ is
$$x : A \mid * : \mathtt{1} \ .$$
This morphism is unique because
$$\Gamma \mid t = \star : \mathtt{1}$$
is an axiom for the terms of unit type $\mathtt{1}$.

- The product of objects A and B is the type $A \times B$. The first and the second projections are the terms
$$p : A \times B \mid \mathtt{fst}\, p : A \ , \qquad\qquad p : A \times B \mid \mathtt{snd}\, p : B \ .$$
Given morphisms
$$z : C \mid t : A \ , \qquad\qquad z : C \mid u : B \ ,$$
the term
$$z : C \mid \langle t, u \rangle : A \times B$$
represents the unique morphism satisfying
$$z : C \mid \mathtt{fst}\, \langle t, u \rangle = t : A \ , \qquad\qquad z : C \mid \mathtt{snd}\, \langle t, u \rangle = u : B \ .$$
Indeed, if $\mathtt{fst}\, s = t$ and $\mathtt{snd}\, s = u$ then
$$s = \langle \mathtt{fst}\, s, \mathtt{snd}\, s \rangle = \langle t, u \rangle \ .$$

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

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

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

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

  Showing that this is the transpose of t amounts to

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

  which is a valid chain of equations in λ-calculus. The transpose is unique, because any morphism $z : C \mid s : A \to B$ that satisfies

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

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

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

  Therefore,

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

  as required.

□

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

Lemma 3.9.2. *A* model $[\![-]\!]$ *of a λ-theory $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$ determines a cartesian closed functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ with*

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

for all basic types A and basic constants c. Moreover, M is unique up to a unique isomorphism of CCC functors, in the sense that given another model N satisfying (3.28)*, there is a unique natural iso $M \cong N$ determined inductively by the comparison maps $M(1) \cong N(1)$,*

$$M(A \times B) \;\cong\; MA \times MB \;\cong\; NA \times NB \;\cong\; N(A \times B)\,,$$

and similarly for $M(B^A)$.

Proof. Straightforward. □

We now have the usual functorial semantics theorem:

Theorem 3.9.3. *For any λ-theory $\mathbb{T}$, the syntactic category $\mathcal{C}_{\mathbb{T}}$ classifies $\mathbb{T}$-models, in the sense that for any cartesian closed category $\mathcal{C}$ there is an equivalence of categories*

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

naturally in $\mathcal{C}$.

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

We can now proceed just as we did in the case of algebraic theories and prove that the semantics of λ-theories in cartesian closed categories is *complete*, in virtue of the syntactic construction of the classifying category $\mathcal{C}_{\mathbb{T}}$. Specifically, a λ-theory $\mathbb{T}$ has a canonical interpretation $[-]$ in the syntactic category $\mathcal{C}_{\mathbb{T}}$, which interprets a basic type A as itself, and a basic constant c of type A as the morphism $[x : \mathtt{1} \mid c : A]$. The canonical interpretation is a model of $\mathbb{T}$, also known as the *syntactic model*, in virtue of the definition of the equivalence relation $[-]$ on terms. In fact, it is a *logically generic* model of $\mathbb{T}$, because by the construction of $\mathcal{C}_{\mathbb{T}}$, for any terms $\Gamma \mid u : A$ and $\Gamma \mid t : A$, we have

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

For the record, we therefore have shown:

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

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

Of course, the syntactic model $[-]$ is the one associated under (3.29) to the identity functor $\mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{T}}$, i.e. it is the *universal* one. It therefore satisfies an equation just in case the equation holds in all models, by the classifying property of $\mathcal{C}_{\mathbb{T}}$, and the preservation of satisfaction of equations by CCC functors (Proposition 3.8.1).

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

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

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

3.10 The internal language of a CCC

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

1. For each basic type A in $\mathbb{T}$ a type τA in $\mathbb{U}$. The translation is then extended to all types by the rules
$$\tau \mathsf{1} = \mathsf{1}\,, \qquad \tau(A \times B) = \tau A \times \tau B\,, \qquad \tau(A \to B) = \tau A \to \tau B\,.$$
2. For each basic constant c of type A in $\mathbb{A}$ a term τc of type τA in $\mathbb{U}$. The translation of terms is then extended to all terms by the rules
$$\begin{aligned} \tau(\mathtt{fst}\,t) &= \mathtt{fst}\,(\tau t)\,, & \tau(\mathtt{snd}\,t) &= \mathtt{snd}\,(\tau t)\,,\\ \tau\langle t, u\rangle &= \langle \tau t, \tau u\rangle\,, & \tau(\lambda x : A\,.\,t) &= \lambda x : \tau A\,.\,\tau t\,,\\ \tau(t\,u) &= (\tau t)(\tau u)\,, & \tau x &= x \quad \text{(if } x \text{ is a variable)}\,. \end{aligned}$$

A context $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ is translated by τ to the context
$$\tau\Gamma = x_1 : \tau A_1, \ldots, x_n : \tau A_n\,.$$
Furthermore, a translation is required to preserve the axioms of $\mathbb{T}$: if $\Gamma \mid t = u : A$ is an axiom of $\mathbb{T}$ then $\mathbb{U}$ proves $\tau\Gamma \mid \tau t = \tau u : \tau A$. It then follows that all equations proved by $\mathbb{T}$ are translated to valid equations in $\mathbb{U}$.

A moment's consideration shows that a translation $\tau : \mathbb{T} \to \mathbb{U}$ is the same thing as a model of $\mathbb{T}$ in $\mathcal{C}_{\mathbb{U}}$, despite being specified entirely syntactically. Clearly, λ-theories and translations between them form a category. Translations compose as functions, therefore composition is associative. The identity translation $\iota_{\mathbb{T}} : \mathbb{T} \to \mathbb{T}$ translates every type to itself and every constant to itself. It corresponds to the canonical interpretation of $\mathbb{T}$ in $\mathcal{C}_{\mathbb{T}}$.

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

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

1. For every object $A \in \mathcal{C}$ there is a basic type $\ulcorner A \urcorner$.
2. For every morphism $f : A \to B$ there is a basic constant $\ulcorner f \urcorner$ whose type is $\ulcorner A \urcorner \to \ulcorner B \urcorner$.
3. For every $A \in \mathcal{C}$ there is an axiom
$$x : \ulcorner A \urcorner \mid \ulcorner \mathsf{1}_A \urcorner x = x : \ulcorner A \urcorner\,.$$

4. For all morphisms $f : A \to B$, $g : B \to C$, and $h : A \to C$ such that $h = g \circ f$, there is an axiom
$$x : \ulcorner A \urcorner \mid \ulcorner h \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner C \urcorner \,.$$

5. There is a constant
$$\mathtt{T} : \mathtt{1} \to \ulcorner 1 \urcorner \,,$$
and for all $A, B \in \mathcal{C}$ there are constants
$$\mathtt{P}_{A,B} : \ulcorner A \urcorner \times \ulcorner B \urcorner \to \ulcorner A \times B \urcorner \,, \qquad \mathtt{E}_{A,B} : (\ulcorner A \urcorner \to \ulcorner B \urcorner) \to \ulcorner B^A \urcorner \,.$$
They satisfy the following axioms:
$$\begin{aligned} u : \ulcorner 1 \urcorner &\mid \mathtt{T} * = u : \ulcorner 1 \urcorner \\ z : \ulcorner A \times B \urcorner &\mid \mathtt{P}_{A,B} \langle \ulcorner \pi_0 \urcorner z, \ulcorner \pi_1 \urcorner z \rangle = z : \ulcorner A \times B \urcorner \\ w : \ulcorner A \urcorner \times \ulcorner B \urcorner &\mid \langle \ulcorner \pi_0 \urcorner (\mathtt{P}_{A,B} w), \ulcorner \pi_1 \urcorner (\mathtt{P}_{A,B} w) \rangle = w : \ulcorner A \urcorner \times \ulcorner B \urcorner \\ f : \ulcorner B^A \urcorner &\mid \mathtt{E}_{A,B} (\lambda x : \ulcorner A \urcorner \,. (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle f, x \rangle))) = f : \ulcorner B^A \urcorner \\ f : \ulcorner A \urcorner \to \ulcorner B \urcorner &\mid \lambda x : \ulcorner A \urcorner \,. (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle (\mathtt{E}_{A,B} f), x \rangle)) = f : \ulcorner A \urcorner \to \ulcorner B \urcorner \end{aligned}$$

The purpose of the constants $\mathtt{T}$, $\mathtt{P}_{A,B}$, $\mathtt{E}_{A,B}$, and the axioms for them is to ensure the isomorphisms $\ulcorner 1 \urcorner \cong \mathtt{1}$, $\ulcorner A \times B \urcorner \cong \ulcorner A \urcorner \times \ulcorner B \urcorner$, and $\ulcorner B^A \urcorner \cong \ulcorner A \urcorner \to \ulcorner B \urcorner$. Types A and B are said to be *isomorphic* if there are terms
$$x : A \mid t : B \,, \qquad y : B \mid u : A \,,$$
such that $\mathbb{T}$ proves
$$x : A \mid u[t/y] = x : A \,, \qquad y : B \mid t[u/x] = y : B \,.$$
Furthermore, an *equivalence of theories* $\mathbb{T}$ and $\mathbb{U}$ is a pair of translations
$$\begin{tikzcd} \mathbb{T} \arrow[r, bend left, "\tau"] & \mathbb{U} \arrow[l, bend left, "\sigma"] \end{tikzcd}$$
such that, for any type A in $\mathbb{T}$ and any type B in $\mathbb{U}$,
$$\sigma(\tau A) \cong A \,, \qquad \tau(\sigma B) \cong B \,.$$

The assignment $\mathcal{C} \mapsto \mathbb{L}(\mathcal{C})$ extends to a functor
$$\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr} \,,$$
where CCC is the category of small cartesian closed categories and functors between them that preserve finite products and exponentials. Such functors are also called *cartesian closed functors* or *ccc functors*. If $F : \mathcal{C} \to \mathcal{D}$ is a cartesian closed functor then $\mathbb{L}(F) : \mathbb{L}(\mathcal{C}) \to \mathbb{L}(\mathcal{D})$ is the translation given by:

1. A basic type $\ulcorner A \urcorner$ is translated to $\ulcorner FA \urcorner$.
2. A basic constant $\ulcorner f \urcorner$ is translated to $\ulcorner Ff \urcorner$.
3. The basic constants $\mathtt{T}$, $\mathtt{P}_{A,B}$ and $\mathtt{E}_{A,B}$ are translated to $\mathtt{T}$, $\mathtt{P}_{FA,BA}$ and $\mathtt{E}_{FA,FB}$, respectively.

We now have a functor $\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}$. How about the other direction? We already have the construction of syntactic category which maps a λ-theory $\mathbb{T}$ to a small cartesian closed category $\mathcal{C}_{\mathbb{T}}$. This extends to a functor

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

because a translation $\tau : \mathbb{T} \to \mathbb{U}$ induces a functor $\mathcal{C}_\tau : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}_{\mathbb{U}}$ in an obvious way: a basic type $A \in \mathcal{C}_{\mathbb{T}}$ is mapped to the object $\tau A \in \mathcal{C}_{\mathbb{U}}$, and a basic constant $x : \mathtt{1} \mid c : A$ is mapped to the morphism $x : \mathtt{1} \mid \tau c : A$. The rest of $\mathcal{C}_\tau$ is defined inductively on the structure of types and terms.

Theorem 3.10.2. *The functors $\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}$ and $\mathcal{C} : \lambda\mathsf{Thr} \to \mathsf{CCC}$ constitute an equivalence of categories, "up to equivalence". This means that for any $\mathcal{C} \in \mathsf{CCC}$ there is an equivalence of catgories*

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

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

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

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

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

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

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

and for all $f : A \to B$ and $g : B \to C$,

$$x : \ulcorner A \urcorner \mid \ulcorner g \circ f \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner C \urcorner .$$

To see that $\eta_{\mathcal{C}}$ is an equivalence of categories, it suffices to show that for every object $X \in \mathcal{C}_{\mathbb{L}(\mathcal{C})}$ there exists an object $\theta_{\mathcal{C}} X \in \mathcal{C}$ such that $\eta_{\mathcal{C}}(\theta_{\mathcal{C}} X) \cong X$. The choice map $\theta_{\mathcal{C}}$ is defined inductively by

$$\theta_{\mathcal{C}} \mathtt{1} = \mathtt{1} , \qquad \theta_{\mathcal{C}} \ulcorner A \urcorner = A ,$$
$$\theta_{\mathcal{C}}(Y \times Z) = \theta_{\mathcal{C}} X \times \theta_{\mathcal{C}} Y , \qquad \theta_{\mathcal{C}}(Y \to Z) = (\theta_{\mathcal{C}} Z)^{\theta_{\mathcal{C}} Y} .$$

We skip the verification that $\eta_{\mathcal{C}}(\theta_{\mathcal{C}} X) \cong X$. In fact, $\theta_{\mathcal{C}}$ can be extended to a functor $\theta_{\mathcal{C}} : \mathcal{C}_{\mathbb{L}(\mathcal{C})} \to \mathcal{C}$ so that $\theta_{\mathcal{C}} \circ \eta_{\mathcal{C}} \cong 1_{\mathcal{C}}$ and $\eta_{\mathcal{C}} \circ \theta_{\mathcal{C}} \cong 1_{\mathcal{C}_{\mathbb{L}(\mathcal{C})}}$.

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

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

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

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

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

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

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

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

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

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

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

1. If $X = \mathtt{1}$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \mathtt{1}) = \mathtt{1}$.

2. If $X = \ulcorner A \urcorner$ is a basic type then A is a type in $\mathbb{T}$. We proceed by induction on the structure of A:

   (a) If $A = \mathtt{1}$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner \mathtt{1} \urcorner) = \mathtt{1}$. The types $\mathtt{1}$ and $\ulcorner \mathtt{1} \urcorner$ are isomorphic via the constant $\mathtt{T} : \mathtt{1} \to \ulcorner \mathtt{1} \urcorner$.

   (b) If A is a basic type then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner A \urcorner) = \ulcorner A \urcorner$.

   (c) If $A = B \times C$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} \ulcorner B \times C \urcorner) = \ulcorner B \urcorner \times \ulcorner C \urcorner$. But we know $\ulcorner B \urcorner \times \ulcorner C \urcorner \cong \ulcorner B \times C \urcorner$ via the constant $\mathtt{P}_{A,B}$.

   (d) The case $A = B \to C$ is similar.

3. If $X = Y \times Z$ then $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}}(Y \times Z)) = \tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Y) \times \tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Z)$. By induction hypothesis, $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Y) \cong Y$ and $\tau_{\mathbb{T}}(\sigma_{\mathbb{T}} Z) \cong Z$, from which we easily obtain

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

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

□

Exercise 3.10.3. In the previous proof we defined, for each $\mathcal{C} \in \mathsf{CCC}$, a functor $\eta_C : \mathcal{C} \to \mathcal{C}_{\mathbb{L}(\mathcal{C})}$. Verify that this determines a natural transformation $\eta : 1_{\mathsf{CCC}} \Longrightarrow \mathcal{C} \circ \mathbb{L}$. Can you say anything about naturality of the translations $\tau_{\mathbb{T}}$ and $\sigma_{\mathbb{T}}$? What would it even mean for a translation to be natural?

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

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

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

3.11 Embedding and completeness theorems

We have considered the λ-calculus as a common generalization of both propositional logic, modelled by poset CCCs such as Boolean and Heyting algebras, and equational logic, modelled by finite product categories. Accordingly, there are then two different notions of "provability", as discussied in Remark 3.8.2; namely the derivability of a closed term $\vdash a : A$, and the derivability of an equation between two (not necessarily closed) terms of the same type $\Gamma \vdash s = t : A$. With respect to the semantics, there are then two different corresponding notions of soundness and completeness: for "inhabitation" of types, and for equality of terms. We consider special cases of these notions in more detail below.

Conservativity

With regard to the former notion, inhabitation, one can also consider the question of how it compares with simple provability in *propositional logic*: e.g. a positive propositional formula ϕ in the variables $p_1, p_2, ..., p_n$ obviously determines a type Φ in the corresponding λ-theory $\mathbb{T}(X_1, X_2, ..., X_n)$ over n basic type symbols. What is the relationship between provability in positive propositional logic, $\mathsf{PPL} \vdash \phi$, and inhabitation in the associated λ-theory, $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$? Let us call this the question of *conservativity* of λ-calculus over PPL. According to the basic idea of the Curry-Howard correspondence from Section 3.1, the λ-calculus is essentially the "proof theory of PPL". So one should expect that starting from an inhabited type Φ, a derivation of a term $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$ should result in a corresponding proof of ϕ in PPL just by "rubbing out the proof terms". Conversely, given a provable formula $\vdash \phi$, one should be able to annotate a proof of it in PPL to obtain a derivation of a term $\mathbb{T}(X_1, X_2, ..., X_n) \vdash t : \Phi$ in the λ-calculus (although perhaps not the same term that one started with, if the proof was obtained from rubbing out a term).

We can make this idea precise semantically as follows. Write $|\mathcal{C}|$ for the poset reflection of a category $\mathcal{C}$, that is, the left adjoint to the inclusion $i : \mathsf{Pos} \hookrightarrow \mathsf{Cat}$, and let $\eta : \mathcal{C} \to |\mathcal{C}|$ be the unit of the adjunction.

Lemma 3.11.1. *If $\mathcal{C}$ is cartesian closed, then so is $|\mathcal{C}|$, and $\eta : \mathcal{C} \to |\mathcal{C}|$ preserves the CCC structure.*

Proof. Exercise! □

Exercise 3.11.2. Prove Lemma 3.11.1.

Corollary 3.11.3. *The syntactic category $\mathsf{PPC}(p_1, p_2, ..., p_n)$ of the positive propositional calculus on n propositional variables is the poset reflection the syntactic category $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$ of the λ-theory $\mathbb{T}(X_1, X_2, ..., X_n)$,*

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

Proof. We already know that $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$ is the free cartesian closed category on n generating objects, and that $\mathsf{PPC}(p_1, p_2, ..., p_n)$ is the free cartesian closed poset on n generating elements. We have an obvious CCC map

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

taking generators to generators, and it extends along the quotient map to $|\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}|$ by the universal property of the poset reflection. Thus it suffices to show that the quotient map preserves, and indeed creates, the CCC structure on $|\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}|$, which follows from the Lemma 3.11.1. □

Remark 3.11.4. Corollary 3.11.3 can be extended to other systems of type theory and logic, with further operations such as CCCs with sums $0, A + B$ ("bicartesian closed categories"), and the full intuitionistic propositional calculus IPC with the logical operations $\bot$ and $p \vee q$. We leave this as a topic for the interested student.

Completeness

As was the case for algebraic and propositional logics, the fact that there is a generic model (Proposition 3.9.4) allows the general completeness theorem stated in Corollary 3.9.5 to be specialized to various classes of special models, via embedding (or "representation") theorems, this time for CCCs, rather than for finite product categories or Boolean/Heyting algebras. We shall consider three such cases: "variable" models, topological models, and Kripke models. Note that this follows that same pattern that we saw for the "proof irrelevant" case of propostional logic, but in some cases, the proofs require much more sophisticated methods.

Variable models

By a *variable model* of the λ-calculus we mean one in a ccc of the form $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, i.e. presheaves on a category $\mathbb{C}$. We regard such a model as "varying over $\mathbb{C}$", just as a presheaf of groups on the simplex category Δ may be seen both as a simplicial group – a simplicial object in the category of groups – and as a group in the category $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ of simplicial sets. The basic fact that we use in specializing Proposition 3.9.4 to such variable models is the following, which is one of the fundamental facts of categorical semantics.

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

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

preserves cartesian closed structure.

Proof. We just evaluate $\mathsf{y}A(X) = \mathbb{C}(X, A)$. It is clear that $\mathsf{y}1(X) = \mathbb{C}(X, 1) \cong 1$ naturally in X, and that $\mathsf{y}(A \times B)(X) = \mathbb{C}(X, A \times B) \cong \mathbb{C}(X, A) \times \mathbb{C}(X, B) \cong (\mathsf{y}A \times \mathsf{y}B)(X)$ for all A, B, X, naturally in all three arguments. For $B^A \in \mathbb{C}$, we then have

$$\mathsf{y}(B^A)(X) = \mathbb{C}(X, B^A) \cong \mathbb{C}(X \times A, B) \cong \widehat{\mathbb{C}}(\mathsf{y}(X \times A), \mathsf{y}B) \cong \widehat{\mathbb{C}}(\mathsf{y}X \times \mathsf{y}A, \mathsf{y}B),$$

since y is full and faithful and preserves $\times$. But now recall that the exponential Q^P of presheaves P, Q is defined at X by the specification

$$Q^P(X) \;=\; \widehat{\mathbb{C}}(\mathsf{y}X \times P, Q)\,.$$

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

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

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

2. *For any terms $\Gamma \mid s, t : A$,*
$$\mathbb{T} \vdash (\Gamma \mid s = t : A)$$
if, and only if,
$$[\![\Gamma \vdash s : A]\!] = [\![\Gamma \vdash t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$$
for every such presheaf model.

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

Topological models

See [Awo00]

Kripke models

See [AR11]

Models based on computability and continuity

See [?]

3.12 Modal operators and monads

See [AK08]

Bibliography

[AK08] Steve Awodey and Kohei Kishida. Topology and modality: Topological semantics for first-order modal logic. *The Review of Symbolic Logic*, 1:146–166, 2008.

[AR11] S. Awodey and F. Rabe. Kripke semantics for Martin-Löf's extensional type theory. *Logical Methods in Computer Science*, 7(3):1–25, 2011.

[Awo00] Steve Awodey. Topological representation of the λ-calculus. *Mathematical Structures in Computer Science*, 10:81–96, 2000.

[Joh82] P.T. Johnstone. *Stone Spaces*. Number 3 in Cambridge studies in advanced mathematics. Cambridge University Press, 1982.

[MH92] Michael Makkai and Victor Harnik. Lambek's categorical proof theory and L auchli's abstract realizability. *Journal of Symbolic Logic*, 57(1):200–230, 1992.

[MR95] Michael Makkai and Gonzalo Reyes. Completeness results for intuitionistic and modal logic in a categorical setting. *Annals of Pure and Applied Logic*, 72:25–101, 1995.