Cambridge Lectures on Categorical Logic and Type Theory

Based on:

An Introduction to Categorical Logic and Type Theory

Steve Awodey Andrej Bauer

[DRAFT: February 19, 2026]

Contents

Part II

Type Theory

Chapter 1

Introduction to Type Theory

1.1 A little history

We begin with a few historical remarks, intended to correct a common misconception about the origins of type theory. For an excellent survey of the history of modern type theory see [Coq22].

The history of type theory is closely tied to the history of logic, to which it is closely related, but distinct. Modern logic emerged together with modern algebra in the 19th century, as something like the algebra of "propositions", as opposed to that of numbers or quantities. As emphasized by Frege, the distinctive feature of logic was the notion of *truth*, which establishes its connection to language, thought, judgement, and other anthropocentric notions—as opposed to, say, numbers and sets, which could be regarded as existing independently of human activity. Although Frege himself strove to establish logic as an objective science, he struggled to define its basic objects in a way that did not rely on their symbolic (one would now say "syntactic") representations. *Objects* were determined as things that could be *named*, possibly by complex symbolic expressions. The basic notion of a *function* was, for Frege, something that derived from a complex name for an object by allowing a constituent name to be replaced by another (think of forming an expression for a polynomial function from an algebraic expression for a number). In this way, the basis of logic was tied to the relation between symbolic expressions and their *meaning* (German *Bedeutung*).

In addition to names of objects, and functional expressions regarded as fragmentary or incomplete names, there were sentences, which were treated simply as names for objects of a special kind, for which Frege coined the term *truth value* (German: *Wahrheitswert*). Functions whose values were truth values – whose expressions were therefore fragmentary sentences (i.e. predicates, or formulas with variables) – were called *concepts* (German: *Begriffe*). In this way, Frege's system of logic was based on (i) objects, including truth values, and functions, including concepts, all of which were in the objective realm of "things", and (ii) their symbolic or linguistic expressions, which were regarded as being in a separate realm. Mediating between the two realms was a third one called that of "senses" (German:

	Symbols	Senses	Things
Total	names, sentences	propositions	objects, truth values
Partial	formulas, predicates	properties	functions, concepts

Table 1.1: Frege's logical inventory

Sinne), which might now be called the *intensions* and included things like *propositions*, which Frege called "thoughts" (German: *Gedanken*).

In this way, certain kinds of *things* (like functions) were inferred, or assumed, to correspond to certain kinds of *expressions*. Accordingly, there were also assumed to be higher-order functions, which resulted from complex expressions by removing the expressions for functions and allowing these arguments to vary (a quantifier is an example of such a higher function). Frege introduced a systematic use of different kinds of variables, notation for variable binding, and other devices to permit the correct formal manipulation of expressions denoting not only objects, but also functions of several arguments, functions of functions, etc. He insisted, moreover, on a strict regimentation of the entities denoted by such expressions. A function of functions could no more take an object as argument than could a noun replace a verb to make a sentence. In this way, Frege's universe of logical entities was partitioned into a rigid hierarchy of disjoint kinds that Russell later called *logical types*.

To be a bit more precise, if we let o denote the type of all objects (including truth values), and $A \to B$ the type of functions from type A to type B, then Frege's system of all logical types T is generated simply by the rules:

$$T ::= \ o \mid (T_1, \ldots, T_n) \to o$$

where $(T_1, \ldots, T_n) \to o$ represents the type of o-valued functions in several arguments of types $T_1, \ldots, T_n$, respectively. (This display can be read as a recursive specification as follows: o is a type; if $T_1, \ldots, T_n$ are types, then $(T_1, \ldots, T_n) \to o$ is a type; and nothing else is a type.)

According to Russell [**?**], a *type* may be defined as "the range of significance of a *propositional function*", which corresponds roughly to Frege's partition of all functions according to what arguments they can take, with the ones taking no arguments being the objects. Whereas for Frege, the values of such functions were arbitrary objects, for Russell they were restricted to being *propositions*, thus corresponding (roughly[1]) to Frege's concepts. Representing the type of propositions by p, Russell's hierarchy of logical types is then generated by the similar rules:

$$T ::= \ o \mid p \mid (T_1, \ldots, T_n) \to p$$

which is to say, all (higher-order) relations on objects and propositions. Of course, Russell did not rest with this, but also introduced a further "ramification", determined by the

[1]We are suppressing the subtlety that Russell's functions were *intensional*, at least in the original formulation, and so took values in propositions, rather than truth values.

quantificational structure of the expression specifying a given (propositional) function. This was despite the fact that the unramified theory already sufficed to block the particular inconsistency in Frege's system that Russell had discovered [**?**]; nonetheless, Russell was apparently worried that other inconsistencies might yet result without the more elaborate ramified theory of types. Moreover, he believed that any such restrictions should arise from a positive theory, and not merely as a way of blocking the known paradoxes; why Frege's considerations regarding the nature of logical objects and functions were not considered adequate for this purpose, I do not know.

The main point if these remarks is that Frege's type theory, which roughly agrees with what is now called *simple type theory*, was in fact *not* motivated by Russell's discovery of an inconsistency, but rather by a principled consideration of the nature of functions and their relationship to the symbolic expressions by which they can be determined. Unfortunately, Frege later effectively violated those restrictions by introducing the notion of the *extension* (German: *Wertverlauf*) of a function, to play the role of a proxy *object*. For ϕ a concept, the extension $\hat{x}\phi$ was essentially the *set* of objects satisfying the concept (the extension of a general function was its "course of values", something like its "graph"). It was this assumption of extensions that led to the inconsistency in his system, via the infamous Law V, which in modern predicate logical notation reads innocently enough as an "axiom of extensionality",

$$\hat{x}\phi = \hat{x}\psi \;\Leftrightarrow\; \forall x(\phi = \psi). \tag{V}$$

It is indeed ironic that Frege, who first formulated and insisted on the natural rules of logical types, ultimately fell victim to violating those very rules.

1.2 Proof relevance

An important aspect of Frege's logic that distinguished it from the algebraic tradition of the time was his emphasis on a rigorous formal system of *derivations*, specified by *rules of inference* that made reference only to the outward logical ("syntactic") form of the expressions, and not to what they were assumed to mean. Frege regarded such formal "gap-free" proofs as essential in determining the logical character of a judgement, unlike some later logicians who regarded the logical status of a judgement as a property of the expression alone—independent of the means of its proof—possibly determined by consideration of external "semantic" interpretations. The *constructive* tradition in logic and foundations can be seen as descending from Frege's invention of a formal deductive system for determining the truth of a judgement, and his insistence on the importance of this notion; the related idea of *proof-relevance* in constructive logic and modern theoretical computer science is arguably further evidence for the correctness of his point of view. The interaction between logic (with its emphasis on *truth*) and type theory (which emphasizes *proofs*) is encapsulated in constructive logic and type theory by the *Curry-Howard correspondence*. In these notes, we shall attempt to highlight this relationship from yet another point of view: the interaction between conventional algebraic structures and what we shall call *categorified* algebraic structures.

In a bit more detail:

- The *Curry-Howard correspondence* is sometimes presented as a somewhat mysterious connection between (say, propositional) logic and (simple) type theory, according to which the "meaning" of a propositional formula is not just a truth-value (or a truth table of values), but rather the collection of all of its proofs. The Propositions-as-Types/Proofs-as-Terms (or Proofs-as-Programs) paradigm is then a proof-theoretic (or computational) alternative to Tarskian, truth-value semantics. The same correspondence also extends to first-order logic and dependent type theory, as we shall see below.

- The algebraic/categorical version of this correspondence is then as follows: not only does propositional *logic* interpret into Boolean and Heyting algebras (and first- and higher-order logic in (pre)toposes), but we also have categorical semantics of the associated *type theories*, like the λ-calculus which is modeled by (locally) cartesian closed categories, such as categories of (pre)sheaves.

- The *poset* algebra of truth-values used for the semantics of propositional or predicate logic (e.g. the Boolean algebra $\{0, 1\}$) is seen to be the *poset reflection* of a suitably structured, proper *category* (e.g. Set), which models a type theory that, in turn, is the "proof-relevant" version of the logic. The general scheme can be represented as follows, with the upper row being the proof-relevant version of the lower one:

Type Theory	Category
Logic	Algebra

  Indeed, a third axis could be added for propositional versus predicate logic and simple versus dependent type theories:

Logic	Algebra	Type Theory	Category
Propositional	Boolean algebra	Simple	CCC
Predicate	Boolean category	Dependent	LCCC

- The relationship between *validity* and *provability*, which is classically described by the relationship between logic and type theory, is described categorically by the (adjoint) notions of categorical generalization and "poset reflection" between (structured) posets and categories. In this way, the Curry-Howard correspondence relates to the idea of "categorification": a structured category whose poset reflection is a given structured poset. For example, a category with finite products is a categorification of a $\wedge$-semilattice, and the category Set is a categorification of the Boolean algebra $\{0, 1\}$.

Finally, some new ideas have recently deepened this perspective: the original Curry-Howard paradigm, relating truth-value semantics (model theory) with type-theoretic syntax (proof theory), has turned out to capture only the first two levels of an infinite hierarchy

of levels of structure. These levels are not merely cumulative, but are related by inclusion, truncation, (co-)reflection, and other operations. The importance of "proof-relevance" that underlies the Propositions-as-Types idea is essentially just one special case of the *coherence* issue that arises everywhere in higher category theory. And the once-bold replacement of *truth-values* and *sets* by *types* in constructive logic and the foundations of computation parallels the replacement of discrete structures (sheaves) by "higher" ones (stacks) in algebra and geometry—except that we have now learned that the gap between the levels is not just a single step, but rather an infinite hierarchy of levels of structure, each just as significant as the first step.

These insights are reflected in current categorical logic in the recent extension from algebraic logic (level 0) and topos theory (level 1) to higher topos theory and homotopy type theory (level ∞). The latter are the focus of much current research, and the unification of the various earlier topics that has been achieved already shows how much we have learned about what happens in passing from 0 to 1, by passing from the finite to the infinite.

1.3 The Curry-Howard correspondence

Consider the following natural deduction proof in propositional calculus.

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

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 $\vdash A$ discards such differences in the proofs that witness it. According to the "proof-relevant" point of view, sometimes called *propositions as types*, one retains as relevant some information about the way in which a proposition is proved. This can be done by annotating the proofs with *proof-terms* as they are constructed, as follows:

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

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

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

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.

1.4 Categorification

From the categorical point of view, the relation of deducibility $A \vdash B$ is a mere preorder. The addition of proof terms $x : A \vdash t : B$ results in a *categorification* of this preorder, in the sense that it becomes a "proper" category, the preordered reflection of which is the deducibility preorder. And now a remarkable fact emerges: it is hardly surprising that the

deducibility preorder has, say, finite products $A \wedge B$ or even exponentials $A \Rightarrow B$; but it is *amazing* that the category with proof terms $x : A \vdash t : B$ as arrows also turns out to be a cartesian closed category, and indeed a proper one, with distinct parallel arrows, such as

$$\pi_2(x)(\pi_1(\pi_1(x))) : (A \wedge B) \wedge (A \Rightarrow B) \longrightarrow B,$$
$$\pi_2(\pi_1(x)) : (A \wedge B) \wedge (A \Rightarrow B) \longrightarrow B.$$

This *category of proofs* contains information about the "proof theory" of the propositional calculus, as opposed to its mere relation of deducibility.

When the calculus of proof terms is formulated as a system of *simple type theory*, it admits an alternate interpretation as a formal system of *function abstraction and application*. This dual interpretation of the system of simple type theory—as the proof theory of propositional logic, and as a formal system for manipulating functions—is sometimes also referred to as the "Curry-Howard correspondence" [Sco70, ML84, Tai68]. From the categorical point of view, it expresses an equivalence between two cartesian closed categories: that of *proofs* in propositional logic and that of *terms* in simple type theory, both of which are categorifications of their common preorder reflection, the deducibility preorder of propositional logic (cf. [MH92]).

In the next chapter, we shall consider this remarkable correspondence in more detail, as well as some extensions of the basic case to λ-calculus, respectively cartesian closed categories, with sums, with natural numbers objects, and with modal operators. In the subsequent chapter, it will be seen that this correspondence extends even further to proofs in quantified predicate logic via dependent type theory and locally cartesian closed categories, and far beyond.

Chapter 2

Simple Type Theory

2.1 The λ-calculus

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

$$fx = t\ .$$

Here we gave the name f to the newly formed function, which takes an argument x to the value t. But we could have expressed the same function without giving it a name; this is sometimes written as

$$x \mapsto t\ ,$$

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

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

This operation is called λ-*abstraction*. For example, $\lambda x.\, \lambda y.\, (x + y)$ is the function that maps an argument a to the function $\lambda y.\, (a + y)$, which in turn maps an argument b to the value $a + b$. The variable x is said to be *bound* in the expression $\lambda x.\, t$.

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

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

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

is allowed, whatever it may mean. We will concentrate here on the typed λ-calculus (but see Example 2.1.7 below). In typed λ-calculus every expression has a *type*, and there are rules for forming valid expressions and assigning types. For example, we can only form an application fa when f has, say, type $A \to B$ and a has type A, and then fa will necessarily have type B. The basic judgement that an expression t has a type T is written as

$$t : T$$

and it is one of the primitive notions of type theory (meaning that it is not defined). 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 $t \in A$ (at least to get started).

Simply-typed λ-calculus. We now give a more formal definition of what constitutes a *simply-typed λ-calculus*. First, we are given a collection of *simple types*, which are generated from some *basic types* by formation of product and function types:

$$\begin{aligned} \text{Basic types} \quad & \mathtt{B} ::= \mathtt{B}_0 \mid \mathtt{B}_1 \mid \mathtt{B}_2 \cdots \\ \text{Simple types} \quad & A ::= \mathtt{B} ::= \mathtt{1} \mid A_1 \times A_2 \mid A_1 \to A_2. \end{aligned}$$

When convenient, we may adopt the convention that function types associate to the right,

$$A \to B \to C \;=\; A \to (B \to C)\ .$$

We assume there is a countable set of *variables* $x, y, z, \ldots$ at our disposal. We are also given a set of *basic constants*. The set of *terms* is generated from variables and basic constants by the following grammar:

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

In words, this means:

1. any variable is a term,
2. each basic constant is a term,
3. the constant $*$ is a term, called the *unit*,
4. if s and t are terms then $\langle s, t \rangle$ is a term, called a *pair*,

5. if t is a term then $\mathtt{fst}\, t$ and $\mathtt{snd}\, t$ are terms,

6. if s and t are terms then $s\, t$ is a term, called an *application*,

7. if x is a variable, A is a type, and t is a term, then $\lambda x : A\,.\, t$ is a term, called a *λ-abstraction*.

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

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

A term t is *closed* if all of its variables are bound, so that $\mathsf{FV}(t) = \emptyset$. 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. We may identify contexts under reordering, regarding them as sets rather than sequences. Contexts may be 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 (not necessarily basic):

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

- The type of a variable is determined by the context:

$$\frac{}{\Gamma, x_n : A_n \mid x_n : A_n}$$

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

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

- The typing rules for pairs and projections are:

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

- The typing rules for application and λ-abstraction are:

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

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

$$\Gamma \mid s : A\,, \qquad\qquad \Gamma \mid t : A\,,$$

the judgment that they are equal is written as

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

Note that s and t necessarily have the same type; it does *not* make sense to compare terms of different types. We have the following rules for equations, the effect of which is to make equality between terms into an equivalence relation at each type, and a congruence with respect to all of the operations, just as for algebraic theories:

- Equality is an equivalence relation:

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

- The substitution rule:

$$\frac{\Gamma \mid s = t : A \qquad \Gamma, x : A \mid u = v : B}{\Gamma \mid u[s/x] = v[t/x] : B}$$

- The weakening rule:

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

- Unit type:

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

- Equations for product types:

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

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

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

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

- Equations for function types:

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

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

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

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

  where the substitution $t[u/x]$ is defined as usual (see the Appendix).

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

Simply-typed λ-theories. Apart from the above rules for equality, which are part of the λ-calculus, we might want to impose additional equations between terms. In this case we speak of a λ-*theory*. Thus, a λ-theory $\mathbb{T}$ is given by a set of basic types and a set of basic constants, called the *signature*, and a set of *equations* of the form

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

Note that we can always state the equations equivalently in *closed* form simply by λ-abstracting all the variables in the context Γ.

We summarize the preceding definitions.

Definition 2.1.1. A *(simply-typed) signature* S is given by a set of *basic types* $(\mathtt{B}_i)_{i\in I}$ together with a set of *basic (typed) constants* $(\mathtt{c}_j : C_j)_{j\in J}$,

$$S \;=\; \big((\mathtt{B}_i)_{i\in I},\, (\mathtt{c}_j : C_j)_{j\in J}\big)\,.$$

A simply-typed λ-theory $\mathbb{T} = (S, E)$ is a simply-typed signature S together with a set of equations between closed terms,

$$E = \big(u_k = v_k : A_k\big)_{k\in K}\,.$$

Example 2.1.2. The theory of a group is a simply-typed λ-theory. It has one basic type $\mathtt{G}$ and three basic constants, the unit $\mathtt{e}$, the inverse $\mathtt{i}$, and the group operation $\mathtt{m}$,

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

with the following familiar equations (which we need not give in closed form):

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

Example 2.1.3. More generally, any (Lawvere) algebraic theory $\mathbb{A}$ (as in Chapter **??**) determines a λ-theory $\mathbb{A}^\lambda$. There is one basic type $\mathtt{A}$ and for each operation f of arity k there is a basic constant $\mathtt{f} : \mathtt{A}^k \to \mathtt{A}$, where $\mathtt{A}^k$ is the k-fold product $\mathtt{A} \times \cdots \times \mathtt{A}$. It is understood that $\mathtt{A}^0 = \mathtt{1}$. The terms of $\mathbb{A}$ are translated to corresponding terms of $\mathbb{A}^\lambda$ in a straightforward manner. For every axiom $u = v$ of $\mathbb{A}$ there is a corresponding one in $\mathbb{A}^\lambda$,

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

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

Example 2.1.4. The theory of a directed graph is a simply-typed theory with two basic types, $\mathtt{V}$ for vertices and $\mathtt{E}$ for edges, and two basic constants, source $\mathtt{src}$ and target $\mathtt{trg}$,

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

There are no equations.

Example 2.1.5. The theory of a simplicial set is a simply-typed theory with one basic type $\mathtt{X}_n$ for each natural number n, and the following basic constants, also for each n, and each $0 \leq i \leq n$:

$$\mathtt{d}_i : \mathtt{X}_{n+1} \to \mathtt{X}_n\,, \qquad \mathtt{s}_i : \mathtt{X}_n \to \mathtt{X}_{n+1}\,.$$

The equations are the usual simplicial identities, which are as follows, for all natural numbers i, j:

$$\begin{aligned} \mathtt{d}_i\mathtt{d}_j &= \mathtt{d}_{j-1}\mathtt{d}_i, \quad \text{if } i < j, \\ \mathtt{s}_i\mathtt{s}_j &= \mathtt{s}_{j+1}\mathtt{s}_i, \quad \text{if } i \leq j, \\ \mathtt{d}_i\mathtt{s}_j &= \begin{cases} \mathtt{s}_{j-1}\mathtt{d}_i, & \text{if } i < j, \\ \mathtt{id}, & \text{if } i = j \text{ or } i = j+1, \\ \mathtt{s}_j\mathtt{d}_{i-1}, & \text{if } i > j+1. \end{cases} \end{aligned}$$

Example 2.1.6. An example of a λ-theory found in the theory of programming languages is the mini-programming language *PCF*. It is a theory in simply-typed λ-calculus with a basic type `nat` for natural numbers, and a basic type `bool` of Boolean values,

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

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

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

The equational axioms of PCF are:

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

Example 2.1.7 (D.S. Scott)**.** Another example of a λ-theory is the *theory of a reflexive type*. This theory has one basic type `D` and two constants

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

satisfying the equation

$$f : \mathtt{D} \to \mathtt{D} \mid \mathtt{r}\,(\mathtt{s}\,f) = f : \mathtt{D} \to \mathtt{D} \tag{2.1}$$

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

$$x : \mathtt{D} \mid \mathtt{s}\,(\mathtt{r}\,x) = x : \mathtt{D} \tag{2.2}$$

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

A reflexive type can be used to interpret the untyped λ-calculus into the typed λ-calculus.

Untyped λ-calculus

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

$$t ::= v \mid t_1\, t_2 \mid \lambda x.\, t\ .$$

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:

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

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

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

where again the substitution $t[u/x]$ is defined as usual (see the Appendix).

The untyped λ-calculus can be translated into the theory of a reflexive type from Example 2.1.7. An untyped context Γ is translated to a typed context Γ^* by typing each variable in Γ with the reflexive type $\mathtt{D}$, i.e., a context $x_1, \ldots, x_k$ is translated to $x_1 : \mathtt{D}, \ldots, x_k : \mathtt{D}$. An untyped term t is translated to a typed term t^* as follows:

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

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

$$\Gamma \mid u = t \tag{2.3}$$

is translated to the judgment

$$\Gamma^* \mid u^* = t^* : \mathtt{D}\,. \tag{2.4}$$

Exercise* 2.1.8. (counts as two!) Prove that if equation (2.3) is provable then equation (2.4) is provable as well. Identify precisely at which point in your proof you need to use equations (2.1) and (2.2). Does provability of (2.4) imply provability of (2.3)?

2.2 Cartesian closed categories

We next review of the theory of cartesian closed categories, which will form the basis for the semantics of simple type theory.

Exponentials

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

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

Thus, when equipped with a suitable order, the *set* $\mathsf{Hom}(P, Q)$ becomes an object of Pos.

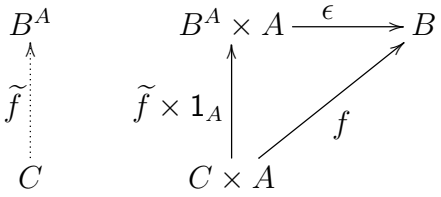
Similarly, given monoids $K, M \in \mathsf{Mon}$, there is a natural monoid structure on the set $\mathsf{Hom}(K, M)$, defined pointwise by

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

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

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

Definition 2.2.1. 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.



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

Definition 2.2.1 is called the *universal property of the exponential*. It is just the category-theoretic way of saying that a function $f : C \times A \to B$ of two variables can be viewed as a function $\widetilde{f} : C \to B^A$ of one variable that maps $z \in C$ to a function $\widetilde{f}z = f\langle z, - \rangle : A \to B$ that maps $x \in A$ to $f\langle z, x \rangle$. The relationship between f and $\widetilde{f}$ is then the expected one:

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

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

In Pos 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, which is easily seen to be monotone. 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*, which is also easily seen to be monotone. We say that the category Pos *has all exponentials*.

Definition 2.2.2. Suppose $\mathcal{C}$ has all finite products. An object $A \in \mathcal{C}$ is *exponentiable* when the exponential B^A exists for every $B \in \mathcal{C}$ (including an associated evaluation map $\epsilon : B^A \times A \to B$). We say that $\mathcal{C}$ *has exponentials* if every object is exponentiable. A *cartesian closed category (ccc)* is a category that has all finite products and exponentials.

Example 2.2.3. Consider again the example of the set $\mathsf{Hom}(M, N)$ of homomorphisms between two monoids M, N, equipped with the pointwise monoid structure. Let $1 = \{u\}$ be the terminal monoid, having only a unit element u. To be a monoid homomorphism, the transpose $\widetilde{h} : 1 \to \mathsf{Hom}(M, N)$ of a homomorphism $h : 1 \times M \to N$ would have to take the unit element $u \in 1$ to the unit homomorphism $u : M \to N$, which is the constant function at the unit $u \in N$. Since $1 \times M \cong M$, that would mean that *all* homomorphisms $h : M \to N$ would have the same transpose, namely $\widetilde{h} = u : 1 \to \mathsf{Hom}(M, N)$. So Mon cannot be cartesian closed. The same argument works in the category Group, and in many related ones.

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

Exercise 2.2.4. Recall that monoids and groups can be regarded as (1-object) categories, and then their homomorphisms are just functors. Thus we have *full* subcategories,

$$\mathsf{Group} \hookrightarrow \mathsf{Mon} \hookrightarrow \mathsf{Cat}\,.$$

Is the category Cat of all (small) categories and functors cartesian closed? What about the subcategory of all *groupoids*,

$$\mathsf{Grpd} \hookrightarrow \mathsf{Cat}\,,$$

defined as those categories in which every arrow is an iso?

Two characterizations of CCCs

Proposition 2.2.5. *In a category $\mathcal{C}$ with binary products an object A is exponentiable if, and only if, the functor*

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

has a right adjoint

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

Proof. If such a right adjoint exists then the exponential of A and B is (B^A, ϵ_B), where $\epsilon_B : B^A \times A \to A$ is the counit of the adjunction at B. Indeed, the universal property of the exponential is just the universal property of the counit $\epsilon : (-)^A \Rightarrow 1_{\mathcal{C}}$.

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

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

as indicated below.

$$\begin{array}{ccc} B^A \times A & \xrightarrow{\epsilon_B} & B \\ {\scriptstyle g^A \times 1_A}\downarrow & & \downarrow{\scriptstyle g} \\ C^A \times A & \xrightarrow[\epsilon_C]{} & C \end{array} \tag{2.5}$$

The counit $\epsilon : -^A \times A \to 1_{\mathcal{C}}$ at B is then ϵ_B itself, and the naturality square for ϵ is then exactly (2.5), 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 adjoints, 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 2.2.6. *A category $\mathcal{C}$ is cartesian closed if, and only if, the following functors all have right adjoints:*

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

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

□

Exercise 2.2.7. Show that being cartesian closed is a *categorical property*, in the sense that it respects equivalence of categories: if $\mathcal{C}$ is cartesian closed and $\mathcal{C} \simeq \mathcal{D}$ then $\mathcal{D}$ is also cartesian closed.

Another consequence of the adjoint formulation is that it implies the possibility of a purely *equational* specification (adjoint structure on a category is "algebraic", in a sense that can be made precise; see [**?**]). It follows that there is a equational formulation of the definition of a cartesian closed category.

Proposition 2.2.8 (Equational version of CCC)**.** *A category $\mathcal{C}$ is cartesian closed if, and only if, it has the following structure:*

1. *An object $\mathsf{1} \in \mathcal{C}$ and a morphism $!_A : A \to \mathsf{1}$ for every $A \in \mathcal{C}$.*
2. *An object $A \times B$ for all $A, B \in \mathcal{C}$ together with morphisms $\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$, and for every pair of morphisms $f : C \to A$, $g : C \to B$ a morphism $\langle f, g \rangle : C \to A \times B$.*
3. *An object B^A for all $A, B \in \mathcal{C}$ together with a morphism $\epsilon : B^A \times A \to B$, and a morphism $\widetilde{f} : C \to B^A$ for every morphism $f : C \times A \to B$.*

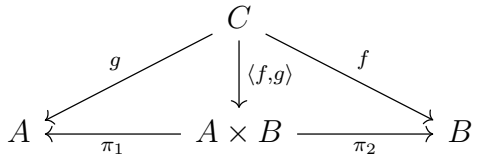
These new objects and morphisms are required to satisfy the following equations:

1. *For every $f : A \to \mathsf{1}$,*
$$f = !_A \ .$$
2. *For all $f : C \to A$, $g : C \to B$, $h : C \to A \times B$,*
$$\pi_1 \circ \langle f, g \rangle = f \ , \qquad \pi_2 \circ \langle f, g \rangle = g \ , \qquad \langle \pi_1 \circ h, \pi_2 \circ h \rangle = h \ .$$
3. *For all $f : C \times A \to B$, $g : C \to B^A$,*
$$\epsilon \circ (\widetilde{f} \times \mathsf{1}_A) = f \ , \qquad (\epsilon \circ (g \times \mathsf{1}_A))^{\sim} = g \ .$$
where for $e : E \to E'$ and $f : F \to F'$ we define
$$e \times f := \langle e\pi_1, f\pi_2 \rangle : E \times F \to E' \times F'.$$

These equations ensure that certain diagrams commute and that the morphisms that are required to exist are unique. For example, let us prove that $(A \times B, \pi_1, \pi_2)$ is the product of A and B. For $f : C \to A$ and $g : C \to B$ we have the morphism $\langle f, g\rangle : C \to A \times B$. The equations

$$\pi_1 \circ \langle f, g\rangle = f \qquad \text{and} \qquad \pi_2 \circ \langle f, g\rangle = g$$

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



Suppose $h : C \to A \times B$ is another morphism such that $f = \pi_1 \circ h$ and $g = \pi_2 \circ h$. Then by the third equation for products we get

$$h = \langle \pi_1 \circ h, \pi_2 \circ h\rangle = \langle f, g\rangle \ ,$$

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

Exercise 2.2.9. Use the equational characterization of CCCs, Proposition 2.2.8, to show that the category Pos of posets and monotone functions *is* cartesian closed, as claimed. Also verify that that Mon is not. Which parts of the definition fail in Mon?

Exercise 2.2.10. Use the equational characterization of CCCs, Proposition 2.2.8, to show that the product category $\Pi_{i\in I}\,\mathcal{C}_i$ of any (set-indexed) family $(\mathcal{C}_i)_{i\in I}$ of cartesian closed categories $\mathcal{C}_i$ is cartesian closed. Is the same true for an arbitrary limit in Cat?

Some proper CCCs

As we have seen **??**, a cartesian closed poset is a $\wedge$-semilattice with exponentials $p \Rightarrow q$, such as a Heyting algebra, or a syntactic category arising from a positive propositional calculus. We next review some important examples of non-poset cartesian closed categories, most of which should be familiar.

Example 2.2.11. 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,

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

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 2.2.12. 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. It also preserves the CCC structure.

Example 2.2.13. The same reasoning as in the previous example shows that the full subcategory $\mathsf{Pos} \hookrightarrow \mathsf{Cat}$ of all small posets and monotone maps is also cartesian closed, and the (limit preserving) inclusion $\mathsf{Pos} \hookrightarrow \mathsf{Cat}$ also preserves exponentials. Note that the (non-full) forgetful functor $U : \mathsf{Pos} \to \mathsf{Set}$ does not, and that $U(Q^P) \subseteq (UQ)^{UP}$ is in general a *proper* subset.

Exercise 2.2.14. Show that there is a full and faithful functor $D : \mathsf{Set} \to \mathsf{Poset}$ that preserves finite limits as well as exponentials. Note the similarity to the example $\mathsf{Grpd} \hookrightarrow \mathsf{Cat}$.

The foregoing examples are instances of the following general situation.

Proposition 2.2.15. *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 2.2.16. *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 the inverse image of the one 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 2.2.17. Which of the foregoing examples follows from which of the previous two propositions?

Example 2.2.18. For any set X, the slice category $\mathsf{Set}/_X$ is cartesian closed. The product of $f : A \to X$ and $g : B \to X$ is the pullback $A \times_X B \to X$, which can be constructed as the set of pairs

$$A \times_X B \to X = \{\langle a, b\rangle \mid fa = gb\}\,.$$

The exponential, however, is *not* simply the set

$$\{h : A \to B \mid f = g \circ h\}\,,$$

(what would the projection to X be?), but rather the set of all pairs

$$\{\langle x, h : A_x \to B_x\rangle \mid x \in X,\ f = g \circ h\}\,,$$

where $A_x = f^{-1}\{x\}$ and $B_x = g^{-1}\{x\}$, with the evident projection to X.

Exercise 2.2.19. Prove that $\mathsf{Set}/_X$ is always cartesian closed. (Hint: Use the fact that $\mathsf{Set}/_X \simeq \mathsf{Set}^X$, and the category of CCCs is closed under products of the underlying categories.)

Lest it be thought that the foregoing example is typical, and every slice of a CCC is again a CCC, one can consider the counterexample of Pos. By an argument like that in [Pal03] for the catesian closed category Grpd of groupoids, the slice categories of Pos need not be cartesian closed.

Exercise 2.2.20. Check that the example given in [Pal03] also works (*mutatis mutandis*) for Pos to show that $\mathsf{Pos}/_X$ is not always cartesian closed.

Example 2.2.21. 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 can use the Yoneda lemma. If Q^P exists, then by Yoneda and the adjunction $(- \times P) \dashv (-^P)$, we would have, for all $c \in \mathbb{C}$,

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

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

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

(This is indeed contravariant in c !) The evaluation morphism $E : Q^P \times P \to Q$ is the natural transformation whose component at c is

$$E_c : \mathsf{Nat}(\mathsf{y}c \times P, Q) \times Pc \to Qc \ ,$$
$$E_c : \langle \eta, x \rangle \mapsto \eta_c \langle \mathsf{1}_c, x \rangle \ .$$

The transpose of a natural transformation $\phi : R \times P \to Q$ is the natural transformation $\widetilde{\phi} : R \to Q^P$ whose component at c is the function that maps $z \in Rc$ to the natural transformation $\widetilde{\phi}_c z : \mathsf{y}c \times P \to Q$, whose component at $b \in \mathcal{C}$ is

$$(\widetilde{\phi}_c z)_b : \mathcal{C}(b, c) \times Pb \to Qb \ ,$$
$$(\widetilde{\phi}_c z)_b : \langle f, y \rangle \mapsto \phi_b \langle (Rf)z, y \rangle \ .$$

Exercise 2.2.22. 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 $\mathsf{Set}^{\to}$, as well as the category of simplicial sets $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ from topology.

Exercise 2.2.23. This exercise is for those 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.

Later in this chapter, we will meet some further examples of CCCs with a more topological flavor:

- Etale spaces over a base space X. This category can be described as consisting of *local homeomorphisms* $f : Y \to X$ and commutative triangles over X between such maps. It is equivalent to the category $\mathsf{Sh}(X)$ of *sheaves* on X (Section 2.8).
- Sheaves for the "+-topology" on a small category $\mathcal{C}$ with (stable) sums $A + B$.
- Dana Scott's category Equ of equilogical spaces (Section 2.8).

2.3 Interpretation of the λ-calculus in a CCC

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

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

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

  (For this purpose, we assume that a CCC structure on $\mathcal{C}$ has been chosen.)

- For every basic constant $\mathtt{c}$ of type C, a morphism $[\![\mathtt{c}]\!] : \mathtt{1} \to [\![\mathtt{C}]\!]$.

The interpretation is then 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,

$$[\![\cdot]\!] = \mathtt{1}\,.$$

- A typing judgment

$$\Gamma \mid t : A$$

  will be interpreted as a morphism

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

  The interpretation is defined inductively by the following rules:

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

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

- A basic constant $\mathtt{c} : \mathtt{C}$ in context Γ is interpreted as the composition

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

- The interpretation of projections and pairs is as follows:

$$\begin{aligned}
[\![\Gamma \mid \langle t, u\rangle : A \times B]\!] &= \langle [\![\Gamma \mid t : A]\!], [\![\Gamma \mid u : B]\!]\rangle : [\![\Gamma]\!] \to [\![A]\!] \times [\![B]\!] \\
[\![\Gamma \mid \mathtt{fst}\, t : A]\!] &= \pi_1 \circ [\![\Gamma \mid t : A \times B]\!] : [\![\Gamma]\!] \to [\![A]\!] \\
[\![\Gamma \mid \mathtt{snd}\, t : A]\!] &= \pi_2 \circ [\![\Gamma \mid t : A \times B]\!] : [\![\Gamma]\!] \to [\![B]\!]\,.
\end{aligned}$$

- The interpretation of application and λ-abstraction is as follows:

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

  where $\epsilon : \llbracket A \to B \rrbracket \times \llbracket A \rrbracket \to \llbracket B \rrbracket$ is the evaluation morphism for $\llbracket B \rrbracket^{\llbracket A \rrbracket}$ and

$$(\llbracket \Gamma, x : A \mid t : B \rrbracket)^{\sim}$$

  is the transpose of the morphism

$$\llbracket \Gamma, x : A \mid t : B \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \to \llbracket B \rrbracket \,.$$

Definition 2.3.1. An interpretation of a λ-theory $\mathbb{T}$ is a *model* of $\mathbb{T}$ if it *satisfies* all the axioms of $\mathbb{T}$, in the sense that for every axiom $\Gamma \mid u = v : A$ of $\mathbb{T}$, the interpretations of u and v coincide as arrows in $\mathcal{C}$,

$$\llbracket \Gamma \mid u : A \rrbracket = \llbracket \Gamma \mid v : A \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket A \rrbracket.$$

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

Proposition 2.3.2 (Soundness)**.** *If $\mathbb{T}$ is a λ-theory and $\llbracket - \rrbracket$ is a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every equation in context $\Gamma \mid s = t : C$ that is provable from the axioms of $\mathbb{T}$, we have*

$$\llbracket \Gamma \mid s : C \rrbracket = \llbracket \Gamma \mid t : C \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket C \rrbracket \,.$$

Briefly, for all $\mathbb{T}$-models $\llbracket - \rrbracket$,

$$\mathbb{T} \vdash (\Gamma \mid s = t : C) \quad \textit{implies} \quad \llbracket - \rrbracket \models (\Gamma \mid s = t : C) \,.$$

The proof is a straightforward induction, first on the typing judgements for the interpretation, and then on the equational rules for the equations. If we stop after the first step, we can consider just the following notion of *inhabitation*.

Remark 2.3.3 (Inhabitation)**.** There is another notion of "provability" for the λ-calculus, related to the Curry-Howard correspondence of section 1.3, relating λ-calculus to the proof theory of propositional logic. If we regard types as "propositions" rather than generalized algebraic structures, and terms as "proofs" rather than operations in such structures, then it is more natural to ask whether there even *is* a term $a : A$ of some type, than whether two terms of the same type are equal $s = t : A$. Of course, this only makes sense when A is considered in the empty context $\cdot \vdash A$, rather than $\Gamma \vdash A$ for non-empty Γ (consider the case where $\Gamma = x : A, \dots$). We say that a type A is *inhabited* (by a closed term) when there is some $\vdash a : A$, and regard an inhabited type A as one that is *provable*. There is then a different notion of soundness related to this notion of provability.

Proposition 2.3.4 (Inhabitation soundness)**.** *If $\mathbb{T}$ is a λ-theory and $[\![-]\!]$ a model of $\mathbb{T}$ in a cartesian closed category $\mathcal{C}$, then for every type A that is inhabited in $\mathbb{T}$, there is a point $1 \to [\![A]\!]$ in $\mathcal{C}$. Thus for all $\mathbb{T}$-models $[\![-]\!]$,*

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

This follows immediately from the fact that $[\![\cdot]\!] = 1$ for the empty context; for then the interpretation of any $\vdash a : A$ is the point

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

Example 2.3.5. 1. A model of an algebraic theory $\mathbb{A}$ (extended to a λ-theory $\mathbb{A}^\lambda$ as in Example 2.1.3) when taken in a CCC $\mathcal{C}$, is just a model of the algebraic theory $\mathbb{A}$ in the underlying finite product category $|\mathcal{C}|_\times$ of $\mathcal{C}$. An important difference, however, is that in defining the *category of models*

$$\mathsf{Mod}_{\mathsf{FP}}(\mathbb{A}, |\mathcal{C}|_\times)$$

we can take *all homomorphisms* of models of $\mathbb{A}$ as arrows, while the arrows in the category

$$\mathsf{Mod}_\lambda(\mathbb{A}^\lambda, \mathcal{C})$$

of λ-models are best taken to be *isomorphisms*, for which one has an obvious way to deal with the contravariance of the function type $[\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}$ (this is discussed in more detail in the next section).

A point to note is that such a model is entirely determined by the interpretation of the *basic* types and terms – *i.e.* the algebra – and the rest of the interpretation is "standard" in the sense that $[\![A \to B]\!] = [\![B]\!]^{[\![A]\!]}$. So in particular, our models are *not* the "Henkin models" that one sometimes sees in the literature.

2. A model of the theory of a reflexive type, Example 2.1.7, in Set must be the one-element set $1 = \{\star\}$ (prove this!). Fortunately, the exponentials in categories of presheaves are *not* computed pointwise; otherwise it would follow that this theory has no non-trivial presheaf models at all! (And then, by Theorem 2.6.6, that the theory itself is degenerate, in the sense that all equations are provable.) That there are non-trivial models is an important fact in the semantics of programming languages and the subject called *domain theory* (see [Sco80b]).

3. A (positive) propositional theory $\mathbb{T}$ may be regarded as a λ-theory, and a model in a cartesian closed poset P is then the same thing as before: an interpretation of the atomic propositions $p_1, p_2, \ldots$ of $\mathbb{T}$ as elements $[\![p_1]\!], [\![p_2]\!], \ldots \in P$, such that the axioms $\phi_1, \phi_2, \ldots$ of $\mathbb{T}$ are all sent to $1 \in P$ by the extension of $[\![-]\!]$ to all formulas,

$$1 = [\![\phi_1]\!] = [\![\phi_2]\!] = \cdots \in P \,.$$

Exercise 2.3.6. How are models of a (not necessarily propositional) λ-theory $\mathbb{T}$ in Cartesian closed *posets* related to models in arbitrary Cartesian closed categories? (*Hint:* Consider the inclusion $\mathsf{CCPos} \hookrightarrow \mathsf{CCC}$. Does it have any adjoints?)

2.4 Functorial semantics

In Chapter **??** we saw how an algebraic theory gives rise to a category with finite products, and its algebras, or models, then correspond to functors preserving finite products on the theory-category. We then arranged the traditional relationship between syntax and semantics into a framework that we called *functorial semantics*. In Chapter **??**, we did the same for propositional logic. As a common generalization of both, the same framework of functorial semantics can be applied to λ-theories and their models in CCCs. The first step is to build the *classifying category* $\mathcal{C}_{\mathbb{T}}$ from a λ-theory $\mathbb{T}$. This is again constructed from the theory itself as a "syntactic" category, as follows:

Definition 2.4.1. For any λ-theory $\mathbb{T}$, the *syntactic category* $\mathcal{C}_{\mathbb{T}}$ is determined as follows.

- The objects of $\mathcal{C}_{\mathbb{T}}$ are the types of $\mathbb{T}$.
- Arrows $A \to B$ are terms in context (of length one):

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

  where two such terms $x : A \mid s : B$ and $x : A \mid s' : B$ are to represent the same morphism when $\mathbb{T}$ proves $x : A \mid s = s' : B$. Note that longer contexts are not required, because we have product types $A_1 \times \cdots \times A_n$.
- Composition of the terms

  $$[x : A \mid s : B] : A \longrightarrow B \qquad \text{and} \qquad [y : B \mid t : C] : B \longrightarrow C$$

  is the term obtained by substituting s for y in t:

  $$[x : A \mid t[s/y] : C] : A \longrightarrow C \ .$$

- The identity morphism on A is the term $[x : A \mid x : A]$ (up to "α-renaming" of variables).

Proposition 2.4.2. *The syntactic category $\mathcal{C}_{\mathbb{T}}$ built from a λ-theory is cartesian closed.*

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

- The terminal object is the unit type $\mathtt{1}$. For any type A the unique morphism $!_A : A \to \mathtt{1}$ is the term

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

  This morphism is indeed unique, because we always have the equation

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

  is an axiom for the terms of unit type $\mathtt{1}$.

- The product of objects A and B is the type $A \times B$. The first and the second projections are the terms

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

  Given morphisms

$$z : C \mid a : A\ , \qquad\qquad z : C \mid b : B\ ,$$

  the term

$$z : C \mid \langle a, b\rangle : A \times B$$

  represents the unique morphism satisfying

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

  Indeed, if $\mathtt{fst}\, t = a$ and $\mathtt{snd}\, t = b$ for some t, then we have

$$t = \langle \mathtt{fst}\, t, \mathtt{snd}\, t\rangle = \langle a, b\rangle\ .$$

  as required.

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

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

  The transpose of a morphism $w : C \times A \mid t : B$ is the term

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

  Showing that this is the transpose of t requires showing, in context $w : C \times A$,

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

  Indeed, we have:

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

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

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

  is equal to $\lambda x : A\,.\,(t[\langle z, x\rangle/w])$, because then

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

  Therefore,

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

  as claimed.

□

The syntactic category $\mathcal{C}_{\mathbb{T}}$ allows us to replace a $\mathbb{T}$-model $[\![-]\!]$ in a CCC $\mathcal{C}$ with a functor $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$. More precisely, we have the following.

Lemma 2.4.3. *A* model $[\![-]\!]$ *of a* λ*-theory* $\mathbb{T}$ *in a cartesian closed category* $\mathcal{C}$ *determines a cartesian closed functor* $M : \mathcal{C}_{\mathbb{T}} \to \mathcal{C}$ *with*

$$M(\mathtt{B}) = [\![\mathtt{B}]\!], \quad M(\mathtt{c}) = [\![\mathtt{c}]\!] : 1 \to [\![C]\!] = M(C)\,, \tag{2.6}$$

for all basic types $\mathtt{B}$ *and basic constants* $\mathtt{c} : C$*. Moreover,* M *is unique up to a unique isomorphism of CCC functors, in the sense that given another model* N *satisfying* (2.6)*, there is a unique natural iso* $M \cong N$*, determined inductively by the comparison maps* $M(1) \cong N(1)$*,*

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

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

Proof. Straightforward structural induction on types and terms with (2.6) as the base case, and using soundness, Proposition 2.3.2, for well-definedness on equivalence classes. Note that the uniqueness up to natural isomorphism uses the fact that all of the morphisms of $\mathcal{C}_{\mathbb{T}}$ are given by terms. □

We then also have the expected functorial semantics theorem:

Theorem 2.4.4. *For any* λ*-theory* $\mathbb{T}$*, the syntactic category* $\mathcal{C}_{\mathbb{T}}$ *classifies* $\mathbb{T}$*-models, in the sense that for any cartesian closed category* $\mathcal{C}$ *there is an equivalence of categories*

$$\mathsf{Mod}_{\lambda}\big(\mathbb{T}, \mathcal{C}\big) \;\simeq\; \mathsf{CCC}\big(\mathcal{C}_{\mathbb{T}}\,, \mathcal{C}\big)\,, \tag{2.7}$$

naturally in $\mathcal{C}$*. The morphisms of* $\mathbb{T}$*-models on the left are the isomorphisms of the underlying structures, and on the right we take the natural isomorphisms of CCC functors.*

Proof. The only thing remaining to show is that, given a model $[\![-]\!]$ in a CCC $\mathcal{C}$ and a CCC functor $f : \mathcal{C} \to \mathcal{D}$, there is an induced model $[\![-]\!]^f$ in $\mathcal{D}$, given by the interpretation $[\![A]\!]^f = f[\![A]\!]$. This is again straightforward, just as for algebraic theories. □

Remark 2.4.5. As mentioned in Example 2.3.5(1) the categories involved in the equivalence (2.7) are *groupoids*, in which every arrow is iso. The reason we have defined them as such is that the contravariant argument A in the function type $A \to B$ prevents us from specifying a non-iso homomorphism of models $h : M \to N$ by the obvious recursion on the type structure.

In more detail, given $h_A : [\![A]\!]^M \to [\![A]\!]^N$ and $h_B : [\![B]\!]^M \to [\![B]\!]^N$, there is no obvious candidate for a map

$$h_{A \to B} : [\![A \to B]\!]^M \longrightarrow [\![A \to B]\!]^N,$$

when all we have are the following induced maps:

$$\begin{array}{ccccc}
[\![A \to B]\!]^M & \xrightarrow{=} & ([\![B]\!]^M)^{[\![A]\!]^M} & \xrightarrow{(h_B)^{[\![A]\!]^M}} & ([\![B]\!]^N)^{[\![A]\!]^M} \\
 & & \uparrow{\scriptstyle ([\![B]\!]^M)^{h_A}} & & \uparrow{\scriptstyle ([\![B]\!]^N)^{h_A}} \\
 & & ([\![B]\!]^M)^{[\![A]\!]^N} & \xrightarrow[(h_B)^{[\![A]\!]^N}]{} & ([\![B]\!]^N)^{[\![A]\!]^N} \xrightarrow{=} [\![A \to B]\!]^N
\end{array}$$

One solution is therefore to take *isos* $h_A : [\![A]\!]^M \cong [\![A]\!]^N$ and $h_B : [\![B]\!]^M \cong [\![B]\!]^N$ and then use the inverses $h_A^{-1} : [\![A]\!]^N \to [\![A]\!]^M$ in the contravariant positions, in order to get things to line up:

$$\begin{array}{ccccc}
[\![A \to B]\!]^M & \xrightarrow{=} & ([\![B]\!]^M)^{[\![A]\!]^M} & \xrightarrow{(h_B)^{[\![A]\!]^M}} & ([\![B]\!]^N)^{[\![A]\!]^M} \\
 & & \downarrow{\scriptstyle ([\![B]\!]^M)^{h_A^{-1}}} \cong & & \cong \downarrow{\scriptstyle ([\![B]\!]^N)^{h_A^{-1}}} \\
 & & ([\![B]\!]^M)^{[\![A]\!]^N} & \xrightarrow[(h_B)^{[\![A]\!]^N}]{} & ([\![B]\!]^N)^{[\![A]\!]^N} \xrightarrow{=} [\![A \to B]\!]^N
\end{array}$$

This suffices to at least get a *category* of models $\mathsf{Mod}_\lambda\big(\mathbb{T}, \mathcal{C}\big)$, rather than just as set, which is enough structure to determine the equivalence (2.7). Note that for an algebraic theory $\mathbb{A}$, this category of λ-models in Set, say, $\mathsf{Mod}_\lambda(\mathbb{A}^\lambda)$ is still the (wide but non-full) subcategory of isomorphisms of conventional (algebraic) $\mathbb{A}$-models

$$\mathsf{Mod}_\lambda(\mathbb{A}^\lambda) \rightarrowtail \mathsf{Mod}(\mathbb{A})\,.$$

We shall consider other solutions to the problem of contravariance below.

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

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

For the record, we therefore have now shown:

Proposition 2.4.6. *For any λ-theory $\mathbb{T}$,*

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

Of course, the syntactic model $[-]$ is the one associated under (2.7) to the identity functor $\mathcal{C}_\mathbb{T} \to \mathcal{C}_\mathbb{T}$, *i.e.* it is the *universal* one. It therefore satisfies an equation just in case the equation holds in *all* models, by the classifying property of $\mathcal{C}_\mathbb{T}$, and the preservation of satisfaction of equations by CCC functors (Proposition 2.3.2).

Corollary 2.4.7 (Completeness)**.** *For any λ-theory $\mathbb{T}$,*

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

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

2.5 The internal language of a CCC

In the case of algebraic theories, we were able to recover the syntactic category from the semantics by taking certain Set-valued functors on the category of models in Set. This then extended to a duality between the category of all algebraic theories and that of all "algebraic categories", which we defined as the categories of Set-valued models of some algebraic theory (and also characterized abstractly). In the (classical) propositional case, this syntax-semantics duality was seen to be exactly the classical Stone duality between the categories of Boolean algebras and of Stone topological spaces. That sort of duality theory seems to be more difficult to formulate for λ-theories, however, now that we have taken the category of models to be just a groupoid (but see Remark **??**). Nonetheless, there is still a correspondence between λ-theories and CCCs, which we get by organizing the former into a category, which is then equivalent to that of the latter. But note that this is analogous to the equivalence between algebraic theories, regarded syntactically, and regarded as finite product categories—rather than to the duality between syntax and semantics.

In order to define the equivalence in question, we first need a suitable notion of *morphism of theories*. A *translation* $\tau : \mathbb{S} \to \mathbb{T}$ of a λ-theory $\mathbb{S}$ into a λ-theory $\mathbb{T}$ is given by the following data:

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

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

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

Definition 2.5.1. Let $\lambda\mathsf{Thr}$ be the category whose objects are λ-theories and morphisms are translations between them.

In this way, we obtain an *isomorphism of sets*,

$$\mathsf{Hom}_{\lambda\mathsf{Thr}}(\mathbb{S}, \mathbb{T}) \cong \mathsf{Mod}_\lambda(\mathbb{S}, \mathcal{C}_\mathbb{T}) \,, \tag{2.8}$$

which is not only natural in $\mathbb{T}$, but also in the theory $\mathbb{S}$, as can be seen by considering the canonical interpretation of $\mathbb{S}$ in $\mathcal{C}_\mathbb{S}$ induced by the identity translation $\iota_\mathbb{S} : \mathbb{S} \to \mathbb{S}$. We can enrich this to an isomorphism of *groupoids* by defining syntactic isomorphisms between translations in $\mathsf{Hom}_{\lambda\mathsf{Thr}}(\mathbb{S}, \mathbb{T})$ in a fairly obvious way so that they correspond bijectively to $\mathbb{S}$-model homomorphisms in $\mathsf{Mod}_\lambda(\mathbb{S}, \mathcal{C}_\mathbb{T})$, which in turn correspond to natural isomorphisms between CCC functors in $\mathsf{Hom}_{\mathsf{CCC}}(\mathcal{C}_\mathbb{S}, \mathcal{C}_\mathbb{T})$, by Theorem 2.4.4,

$$\mathsf{Hom}_{\lambda\mathsf{Thr}}(\mathbb{S}, \mathbb{T}) \cong \mathsf{Mod}_\lambda(\mathbb{S}, \mathcal{C}_\mathbb{T}) \simeq \mathsf{Hom}_{\mathsf{CCC}}(\mathcal{C}_\mathbb{S}, \mathcal{C}_\mathbb{T}) \,.$$

The equivalence $\mathsf{Hom}_{\lambda\mathsf{Thr}}(\mathbb{S}, \mathbb{T}) \simeq \mathsf{Hom}_{\mathsf{CCC}}(\mathcal{C}_\mathbb{S}, \mathcal{C}_\mathbb{T})$ suggests that the functor $\mathcal{C}_{(-)} : \lambda\mathsf{Thr} \to \mathsf{CCC}$ participates in an equivalence of categories,

$$\lambda\mathsf{Thr} \simeq \mathsf{CCC} \,,$$

between λ-theories and cartesian closed categories.

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

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

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

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

5. There is a constant
$$\mathtt{T} : 1 \to \ulcorner \mathtt{1} \urcorner \, ,$$
and for all $A, B \in \mathcal{C}$ there are constants
$$\mathtt{P}_{A,B} : \ulcorner A \urcorner \times \ulcorner B \urcorner \to \ulcorner A \times B \urcorner \, , \qquad \mathtt{E}_{A,B} : (\ulcorner A \urcorner \to \ulcorner B \urcorner) \to \ulcorner B^A \urcorner \, .$$
They satisfy the following axioms:
$$\begin{aligned}
u : \ulcorner \mathtt{1} \urcorner \mid \mathtt{T} * &= u : \ulcorner \mathtt{1} \urcorner \\
z : \ulcorner A \times B \urcorner \mid \mathtt{P}_{A,B} \langle \ulcorner \pi_1 \urcorner z, \ulcorner \pi_2 \urcorner z \rangle &= z : \ulcorner A \times B \urcorner \\
w : \ulcorner A \urcorner \times \ulcorner B \urcorner \mid \langle \ulcorner \pi_1 \urcorner (\mathtt{P}_{A,B} w), \ulcorner \pi_2 \urcorner (\mathtt{P}_{A,B} w) \rangle &= w : \ulcorner A \urcorner \times \ulcorner B \urcorner \\
f : \ulcorner B^A \urcorner \mid \mathtt{E}_{A,B} (\lambda x : \ulcorner A \urcorner \, . \, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle f, x \rangle))) &= f : \ulcorner B^A \urcorner \\
f : \ulcorner A \urcorner \to \ulcorner B \urcorner \mid \lambda x : \ulcorner A \urcorner \, . \, (\ulcorner \mathsf{ev}_{A,B} \urcorner (\mathtt{P}_{A,B} \langle (\mathtt{E}_{A,B} f), x \rangle)) &= f : \ulcorner A \urcorner \to \ulcorner B \urcorner
\end{aligned}$$

The purpose of the constants $\mathtt{T}$, $\mathtt{P}_{A,B}$, $\mathtt{E}_{A,B}$, and the axioms for them is to ensure the isomorphisms $\ulcorner \mathtt{1} \urcorner \cong \mathtt{1}$, $\ulcorner A \times B \urcorner \cong \ulcorner A \urcorner \times \ulcorner B \urcorner$, and $\ulcorner B^A \urcorner \cong \ulcorner A \urcorner \to \ulcorner B \urcorner$. Types A and B are said to be *isomorphic* if there are terms
$$x : A \mid t : B \, , \qquad\qquad y : B \mid u : A \, ,$$
such that $\mathbb{S}$ proves
$$x : A \mid u[t/y] = x : A \, , \qquad\qquad y : B \mid t[u/x] = y : B \, .$$

Furthermore, an *equivalence of theories* $\mathbb{S}$ and $\mathbb{T}$ is a pair of translations
$$\mathbb{S} \underset{\sigma}{\overset{\tau}{\rightleftarrows}} \mathbb{T}$$
such that, for any type A in $\mathbb{S}$ and any type B in $\mathbb{T}$,
$$\sigma(\tau A) \cong A \, , \qquad\qquad \tau(\sigma B) \cong B \, .$$

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

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

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

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

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

Theorem 2.5.2. *The functors $\mathbb{L} : \mathsf{CCC} \to \lambda\mathsf{Thr}$ and $\mathcal{C} : \lambda\mathsf{Thr} \to \mathsf{CCC}$ constitute an equivalence of categories "up to equivalence" (a* biequivalence *of* 2-categories*). This means that for any $\mathcal{C} \in \mathsf{CCC}$ there is an equivalence of categories*

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\tau_{\mathbb{S}} A = \ulcorner A \urcorner \, .$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

□

Composing the isomorphism 2.8 with the equivalence 2.7 we can formulate the foregoing Theorem 2.5.2 as an adjoint equivalence.

Corollary 2.5.3. *There is a biequivalence between the categories* $\lambda\mathsf{Thr}$ *of* λ*-theories and translations between them (and isos thereof), and the category* CCC *of cartesian closed categories and CCC functors (and natural isos),*

$$\begin{aligned}\mathsf{Hom}_{\lambda\mathsf{Thr}}\big(\mathbb{T}, \mathbb{L}\mathcal{C}\big) &\cong \mathsf{Mod}_{\lambda}\big(\mathbb{T}, \mathcal{C}\big)\,, \\ &\simeq\ \mathsf{Hom}_{\mathsf{CCC}}\big(\mathcal{C}_{\mathbb{T}}\,, \mathcal{C}\big)\,.\end{aligned}$$

This is mediated by an adjunction,

$$\mathsf{CCC} \underset{\mathcal{C}}{\overset{\mathbb{L}}{\rightleftarrows}} \lambda\mathsf{Thr}$$

with $\mathcal{C} \dashv \mathbb{L}$*, between the* syntactic category *functor* $\mathcal{C}$ *and the* internal language *functor* $\mathbb{L}$*.*

Exercise 2.5.4. In the proof of Theorem 2.5.2 we defined, for each $\mathcal{C} \in \mathsf{CCC}$, a functor $\eta_C : \mathcal{C} \to \mathcal{C}_{\mathbb{L}(\mathcal{C})}$. Verify that this determines a natural transformation $\eta : \mathbf{1}_{\mathsf{CCC}} \Longrightarrow \mathcal{C} \circ \mathbb{L}$ which is an equivalence of categories. What about the translation $\epsilon_{\mathbb{T}} : \mathbb{T} \to \mathbb{L}(\mathcal{C}_{\mathbb{T}})$—is that an isomorphism?

See the book [LS88] for another approach to the biequivalence of Corollary 2.5.3, which turns it into an equivalence of categories by fixing the CCC structure and requiring it to be preserved *strictly*.

Lawvere's fixed point theorem

As an application of the internal language of a CCC, we can use the λ-calculus to give a neat proof of a fixed point theorem for CCCs due to Lawvere [Law69]. Andrej Bauer has called Lawvere's theorem the "quintessential diagonal argument" [Baub].

Theorem 2.5.5 (Lawvere)**.** *In any cartesian closed category, if a map* $e : A \to B^A$ *is a pointwise surjection, then every map* $f : B \to B$ *has a fixed point.*

By "pointwise surjection" we mean a map that induces a surjection from points $1 \to A$ to points $1 \to B^A$ by composition.

Proof. Given $f : B \to B$, consider the map $\lambda x : A.f(ex)x : 1 \to B^A$. Since e is pointwise surjective, there is a point $a : 1 \to A$ such that $ea = \lambda x : A.f(ex)x$. Thus

$$(ea)a = (\lambda x : A.f(ex)x)a = f(ea)a\,,$$

so $(ea)a : 1 \to B$ is a fixed point of $f : B \to B$. □

Among the consequences of this theorem, as stated in [Law69], are: Cantor's theorem (Corollary 1.2); Gödel's incompleteness theorem (Theorem 3.3); and Tarski's indefinability of truth (Theorem 3.2). These are all derived from the contrapositive form of Lawvere's fixed point theorem 2.5.5: if a certain object B has an endomap with no fixed points, then for no A can there be a pointwise surjection $A \to B^A$. To infer Cantor's theorem, for instance: in Set the contrapositive form of Theorem 2.5.5 implies that there is no pointwise surjection from a set A to its powerset $\mathcal{P}A \cong 2^A$, because the "negation" map $\neg : 2 \to 2$ has no fixed points. (The same argument works in any topos, see [Baua].)

Lawvere's original version is a bit more general, but even in the present form it is clear that Lawvere's fixed point theorem is the essence of many familiar diagonal arguments.

2.6 Embedding theorems and completeness

We have considered the λ-calculus as a common generalization of both propositional logic, modeled by poset CCCs such as Boolean and Heyting algebras, and equational logic, modeled by finite product categories. Accordingly, there are then two different notions of "provability", as discussed in Remark 2.3.3; namely, the derivability of a closed term $\vdash a : A$, and the derivability of an equation between two (not necessarily closed) terms of the same type $\Gamma \vdash s = t : A$. With respect to the semantics, there are then two different corresponding notions of soundness and completeness: for "inhabitation" of types, and for equality of terms. We consider special cases of these notions in more detail below.

Conservativity

With regard to the former notion, inhabitation, one can 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 1.3, 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 2.6.1. *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 2.6.2. Prove Lemma 2.6.1.

Corollary 2.6.3. *The syntactic category $\mathsf{PPC}(p_1, p_2, ..., p_n)$ of the positive propositional calculus on n propositional variables is the poset reflection of the syntactic category $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$ of the λ-theory $\mathbb{T}(X_1, X_2, ..., X_n)$,*

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

Proof. We already know that $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$ 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. From the universal property of $\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}$, we get a 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 $|\mathcal{C}_{\mathbb{T}(X_1,X_2,...,X_n)}|$. But that follows from Lemma 2.6.1. □

Remark 2.6.4. Corollary 2.6.3 can be extended to other systems of type theory and logic, with further operations such as CCCs with sums $0, A+B$ ("bicartesian closed categories"), and the full intuitionistic propositional calculus IPC with the logical operations $\bot$ and $p \vee q$. We leave this as a topic for the interested student.

Completeness

As was the case for equational theories and propositional logic, the fact that there is a generic model (Proposition 2.4.6) allows the general completeness theorem stated in Corollary 2.4.7 to be specialized to various classes of special models, via embedding (or "representation") theorems, this time for CCCs, rather than for finite product categories or Boolean/Heyting algebras. We shall consider three such cases: "variable" models, Kripke models, and topological models. In each case, an "embedding theorem" of the form:

Every CCC embeds into one of the special form $\mathcal{X}$.

gives rise to a completeness theorem of the form:

For all λ-theories $\mathbb{T}$, if $1 \to [\![A]\!]^M$ in all $\mathbb{T}$-models M in all $\mathcal{X}$, then $\mathbb{T} \vdash a : A$,
and if $[\![a]\!]^M = [\![b]\!]^M : 1 \to [\![A]\!]$ in all $\mathbb{T}$-models M in all $\mathcal{X}$, then $\mathbb{T} \vdash a = b : A$.

This of course follows the same pattern that we saw for the simpler "proof relevant" case of equational (*i.e.* finite product) theories, and the even simpler "proof irrelevant" case of propositional logic, but now the proofs of some of the embedding theorems for CCCs require more sophisticated methods.

Variable models

By a *variable model* of the λ-calculus we mean one in a CCC of the form $\widehat{\mathbb{C}} = \mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$, i.e. presheaves on a (small) "index category" $\mathbb{C}$. We regard such a model as "varying over $\mathbb{C}$", just as we saw earlier that a presheaf of groups on *e.g.* the simplex category Δ may be seen both as a simplicial group—a simplicial object in the category of groups—and as a group object in the category $\mathsf{Set}^{\Delta^{\mathsf{op}}}$ of simplicial sets.

The basic embedding theorem that we shall use in specializing Proposition 2.4.6 to such variable models is the following, which is one of the fundamental facts of categorical semantics.

Lemma 2.6.5. *For any small cartesian closed category $\mathbb{C}$, the Yoneda embedding*

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

preserves the cartesian closed structure.

This is of course the "categorified" analogue of Lemma **??**, which we used for the Kripke completeness of the positive propositional calculus PPC.

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

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

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

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

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

For an early version of the following theorem (and much more), see the nice paper [Sco80b] by Dana Scott.

Theorem 2.6.6. *For any λ-theory $\mathbb{T}$, we have the following:*

(i) A type A is inhabited,

$$\mathbb{T} \vdash a : A$$

if, and only if, for every a small category $\mathbb{C}$, in every $\mathbb{T}$-model $[\![-]\!]$ in presheaves $\widehat{\mathbb{C}}$, there is a point

$$1 \to [\![A]\!]\,.$$

(ii) For any terms $\Gamma \mid s, t : A$*,*

$$\mathbb{T} \vdash (\Gamma \mid s = t : A)$$

if, and only if,

$$[\![\Gamma \vdash s : A]\!] = [\![\Gamma \vdash t : A]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$$

for every presheaf model.

Proof. We simply specialize the general completeness statement of Corollary 2.4.7 to CCCs of the form $\widehat{\mathbb{C}}$ using Lemma 2.6.5, together with the fact that the Yoneda embedding is full (and therefore reflects inhabitation) and faithful (and therefore reflects satisfaction of equations). □

Exercise 2.6.7. Show that not every presheaf topos $\mathsf{Set}^{\mathbb{C}^{\mathrm{op}}}$ admits a CCC embedding into a category of the form Set/X for a set X (you may assume the fact that the theory of a reflexive type (Example 2.1.7), is *not* trivial).

2.7 Kripke models

By a Kripke model of (a theory $\mathbb{T}$ in) the λ-calculus, we mean a model $[\![-]\!]$ in the sense of Definition 2.3.1 in a presheaf CCC of the form Set^K for a poset K, *i.e.* a variable model in the sense of the previous section, where the domain of variation is just a *poset*, rather than a proper category. As with Kirpke models of propositional logic, we can regard such a model as varying through (branching) time, over a causally ordered state space, or some other (partially-)ordered parameter space. Note that we use "*covariant* presheaves", i.e. functors $K \to \mathsf{Set}$, to model such variable sets, as is more customary in Kripke semantics. By Theorem 2.4.4, such a model $(K, [\![-]\!])$ is essentially the same thing as a CCC functor $M : \mathcal{C}_{\mathbb{T}} \to \mathsf{Set}^K$, taking values in "variable sets". Regarding the λ-calculus as the proof theory of the propositional calculus via the Curry-Howard correspondence (Section 1.3), it is perhaps not surprising that it should be (inhabitation) *complete* with respect to such Kripke models, in light of Theorem **??**. Completeness with respect to *equations between terms* is entirely another matter, though; while true, the proof is far from a simple generalization of other known results. It can be seen as a verification that the usual notion of $\beta\eta$-equivalence is the "right" notion of equality for proofs.

Example 2.7.1 (Algebraic theories)**.** Before considering such questions, however, let us first spell out explicitly what such a Kripke model looks like for the simple example of a theory $\mathbb{T}$ of an object with a distinguished element, and a commutative binary operation,

$$\mathbb{T} \;=\; \left(\mathtt{B},\ \mathtt{u} : \mathtt{B}, * : \mathtt{B} \times \mathtt{B} \to \mathtt{B},\ \mathtt{x} * \mathtt{y} = \mathtt{y} * \mathtt{x}\right).$$

There is one basic type symbol $\mathtt{B}$, a constant $\mathtt{u} : \mathtt{B}$, a binary operation symbol $* : \mathtt{B} \times \mathtt{B} \to \mathtt{B}$, and a single equation $x, y : \mathtt{B} \,|\, x * y = y * x : \mathtt{B}$.

Let K be a poset with ordering relation $j \leq k$ for $j, k \in K$. Unwinding the general definition for this special case, a *Kripke model* M *of* $\mathbb{T}$ *over* K then consists, first, of a family of sets $(M_k)_{k\in K}$, equipped with functions

$$m_{j,k} : M_j \to M_k \qquad (\text{for all } j \leq k \in K)\,,$$

satisfying the "compatibility conditions":

$$m_{k,k} = 1_{M_k}, \qquad m_{j,k} \circ m_{i,j} = m_{i,k} \qquad (\text{for all } j \leq k \in K)\,.$$

This is of course exactly a functor $M : K \to \mathsf{Set}$, as the interpretation $M = [\![\mathtt{B}]\!]$ of the basic type symbol $\mathtt{B}$. Such a variable set M may be thought of as a "set that is changing through time", in that its elements $m_j \in M_j$ develop and change at different stages $j \leq k \in K$. Note that, while new elements may appear at later stages, and distinct elements may become equal, once present, an element never vanishes, nor do elements ever split apart, because the functions $m_{j,k} : M_j \to M_k$ are of course single-valued.

Next, for each $k \in K$ we have an element

$$u_k : M_k\,,$$

and these should satisfy

$$m_{j,k}(u_j) = u_k \qquad (\text{for all } i \leq j \leq k \in K)\,.$$

That is to say, we have an element or "point" $u : 1 \to M$ of M as a "variable set" in Set^K.

Finally, for all $k \in K$ we need functions

$$s_k : M_k \times M_k \to M_k$$

satisfying

$$m_{j,k}\big(s_j(x,y)\big) \;=\; s_k\big(m_{j,k}(x), m_{j,k}(y)\big) \qquad (\text{for all } j \leq k \in K \text{ and } x, y \in M_j)\,.$$

This is of course just a natural transformation $s : M \times M \to M$, as the interpretation $s = [\![*]\!]$ of the operation symbol $* : \mathtt{B} \times \mathtt{B} \to \mathtt{B}$. The idea is that the $*$-product of two elements changes along with those elements, which one sees more clearly by writing $*$ for s:

$$m_{j,k}(x *_j y) \;=\; m_{j,k}(x) *_k m_{j,k}(y) \qquad (\text{for all } j \leq k \in K \text{ and } x, y \in M_j)\,.$$

In other word, it doesn't matter "when" one takes the $*$-product.

Finally, the interpretation $(M, u, s) = [\![\mathtt{B}, \mathtt{u}, *]\!]$ should satisfy the equation $x, y : \mathtt{B} \,|\, x * y = y * x : \mathtt{B}$, meaning that

$$s_k(x,y) = s_k(y,x) \qquad (\text{for all } k \in K)\,.$$

This is because two natural transformations are equal just if all of their components are equal. Thus, in sum, a Kripke model of this theory $\mathbb{T}$ is just a model of the underlying algebraic theory in the functor category Set^K, which is of course the same thing as a functor from K to the usual category of $\mathbb{T}$-models in Set,

$$\mathsf{Mod}_{\mathbb{T}}(\mathsf{Set}^K) \;=\; \mathsf{Mod}_{\mathbb{T}}(\mathsf{Set})^K\,.$$

Example 2.7.2 (Higher-order theories)**.** A theory involving a "higher-order" operation, such as the section $s : (D \to D) \to D$ in (the theory of) a reflexive type (Example 2.1.7) is no more "non-standard" than an algebraic one, once we recall how function types are interpreted, namely *not pointwise*. Let $D = [\![\mathtt{D}]\!]$ be the interpretation of the basic type $\mathtt{D}$, so that $[\![\mathtt{D} \to \mathtt{D}]\!] = D^D : K \to \mathsf{Set}$ is a presheaf exponential. At each $k \in K$, we then have,

$$(D^D)_k = \mathsf{Set}^K\big(D \times K(k,-), D\big)\,.$$

Now observe that this set is trivial except on the upset $\uparrow k$, because $K(k,j)$ is empty unless $k \leq j$, so that $\mathsf{Set}^K\big(D \times K(k,j), D\big) = 1$ except when $j \in \uparrow k$. On $\uparrow k$, it consists of natural transformations

$$\mathsf{Set}^{\uparrow k}\big(D{\uparrow}k, D{\uparrow}k\big)\,,$$

where $D{\uparrow}k : {\uparrow}k \to \mathsf{Set}$ is D restricted to the upset $\uparrow k \subseteq K$, *i.e.* the composite

$${\uparrow}k \hookrightarrow K \xrightarrow{D} \mathsf{Set}\,.$$

Given any such natural transformation $\vartheta : D{\uparrow}k \longrightarrow D{\uparrow}k$, and any $k \leq j$, the action of the functor,

$$(D^D)_k \to (D^D)_j$$

on ϑ is simply to restrict it further to $\uparrow j \subseteq \uparrow k$, thus taking ϑ to

$$\vartheta{\uparrow}j : D{\uparrow}j \longrightarrow D{\uparrow}j\,.$$

This is just the same function as ϑ, but with the restricted domain of definition $\uparrow j \subseteq \uparrow k$. Note that the effect of this restriction may be to identify elements ϑ, and that not every element defined at j need be the restriction of one defined at k for $j \leq k$, so the transition maps need be neither injective nor surjective.

The section $s : (D \to D) \to D$ therefore takes, at each $k \in K$, such a $\vartheta : D{\uparrow}k \longrightarrow D{\uparrow}k$ to an element $s_k(\vartheta) \in D_k$, respecting the restrictions $\uparrow j \subseteq \uparrow k$ in the sense that

$$d_{k,j}s_k(\vartheta) = s_j(\vartheta{\uparrow}j) \in D_j\,,$$

where $d_{k,j} : D_k \to D_j$ is the action of the functor $D : K \to \mathsf{Set}$.

In this way, the presheaf exponential $D^D : K \to \mathsf{Set}$ is entirely determined by the "base-case" $D : K \to \mathsf{Set}$, and is still a "full function space" at each $k \in K$, but the functorial action in k requires it to not be just $D_k^{D_k}$ (which for a reflexive type would then be trivial at all $k \in K$). Rather, it must take the entire segment $\uparrow k$ into account—much in the way that $k \Vdash \varphi \Rightarrow \psi$ was determined for Kripke models of the intuitionistic propositional calculus IPC by considering all $j \geq k$. (Indeed, one can explicitly formulate the Kripke semantics for simple type theory in the usual Kripke-forcing style $k \Vdash a : A$, *cf.* [AGH24] and Section **??** below.)

The proof of the following completeness theorem relies on a deep result from topos theory (for the proof of which, see [Joh03, §xy]). We state it in the following form:

Theorem 2.7.3 (Joyal-Tierney, [JT84])**.** *For every Grothendieck topos $\mathcal{E}$ there is a localic topos $\mathsf{Sh}(L)$ and a connected, locally connected geometric morphism $c : \mathsf{Sh}(L) \to \mathcal{E}$.*

This theorem implies in particular that, for every small CCC $\mathcal{C}$ there is a poset K and a fully faithful CCC functor $\mathcal{C} \hookrightarrow \mathsf{Set}^K$. From this, the completeness of Kripke semantics then follows easily:

Theorem 2.7.4 (Kripke completeness for λ-calculus)**.** *For any λ-theory $\mathbb{T}$:*

(i) A type A is inhabited just if it has a point $1 \to [\![A]\!]$ in every Kripke model $(K, [\![-]\!])$.

(ii) Two terms are provably equal, $\mathbb{T} \vdash (\Gamma \mid s = t : A)$, just if they are equal in every Kripke model $(K, [\![-]\!])$,

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

In the following chapter, we shall see that this result holds as well for *dependent* type theory with the Σ, Π, and Eq type-formers. For the proof, see [AR11], as well as [AGH24].

Remark 2.7.5 (For readers familiar with topos theory)**.** Let us see how to get from Theorem 2.7.3 to the fact used here, that for every small CCC $\mathcal{C}$ there is a poset K and a fully faithful, CCC functor $\varphi : \mathcal{C} \hookrightarrow \mathsf{Set}^K$. First, compose the Yoneda embedding $\mathsf{y} : \mathcal{C} \hookrightarrow \widehat{\mathcal{C}}$ with the inverse image $c^* : \widehat{\mathcal{C}} \hookrightarrow \mathsf{Sh}(L)$ of the Joyal-Tierney cover, which is also fully faithful and CCC. Then compose further with the inclusion $i_* : \mathsf{Sh}(L) \hookrightarrow \mathsf{Set}^{L^{\mathsf{op}}}$ of sheaves into presheaves, which is also fully faithful and CCC. So we can take $K = L^{\mathsf{op}}$ to get the desired CCC embedding $\varphi = i_* \circ c^* \circ \mathsf{y} : \mathcal{C} \hookrightarrow \mathsf{Set}^K$. See [Awo00, AR11] for more details.

2.8 Topological models

From presheaves to posets to spaces to sheaves.

Posets

Since the category Pos is cartesian closed, we can take models of λ-theories there. Are such poset models sufficient to test for provability? The answer depends in general on the kinds of theories: Plotkin [Plo73] shows that for theories with one basic type, no basic terms, and no equations, the models in the category Set with the base type interpreted as a *finite* set are already sufficient. And Friedman [Fri75] showed that the single model with one countably infinite base type is also sufficient. For theories with basic terms (but still no equations), other results are known for the category Pos; see [Sim95] for a summary.

We shall show here that for *arbitrary* theories, with basic types, basic terms, and equations, there are enough models in the category dopFib of posets and discrete opfibrations, provided these are taken relative to an arbitrary base poset K.

Definition 2.8.1. A *discrete opfibration* of posets is a monotone map $\pi : P \to K$ with the property that, for every $p \in P$ and $\pi p \leq k \in K$, there is a unique $p \leq q \in P$ with $\pi q = k$.

This "lifting property" can be equivalently reformulated as saying that every commutative square as follows has a unique diagonal filler, where $\mathbb{2} = (0 \leq 1)$.

$$\begin{array}{ccc} 1 & \longrightarrow & P \\ {\scriptstyle 0}\downarrow & \nearrow & \downarrow{\scriptstyle \pi} \\ \mathbb{2} & \longrightarrow & K \end{array} \tag{2.9}$$

Lemma 2.8.2. *The (full!) subcategory of all such maps*

$$\mathsf{dopFib}/_K \hookrightarrow \mathsf{Pos}/_K$$

is equivalent to Set^K*. In particular, this category is therefore cartesian closed.*

For the proof, one can consider the universal discrete opfibration (with small fibers) $u : \mathsf{Set}^\bullet \to \mathsf{Set}$ in CAT, the category of large categories. A covariant presheaf $P : K \to \mathsf{Set}$ then fits into a pullback diagram

$$\begin{array}{ccc} \int P & \longrightarrow & \mathsf{Set}^\bullet \\ {\scriptstyle \pi}\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ K & \xrightarrow[P]{} & \mathsf{Set} \end{array}$$

with the *category of elements* $\pi : \int_K P \to K$ on the left, and indeed, every discrete opfibration $p : D \to K$ arises in this way from an essentially unique $P : K \to \mathsf{Set}$, namely the one with $P(k) = p^{-1}(k)$. Moreover, every pullback of a discrete fibration (such as u is a discrete fibration, as is easily seen by considering lifting (2.9).

Exercise 2.8.3. Fill in the details of the proof just sketched that $\mathsf{Set}^K \simeq \mathsf{dopFib}/_K$.

Exercise 2.8.4. Show that the inclusion $\mathsf{dopFib}/_K \hookrightarrow \mathsf{Pos}/_K$ is always full, as claimed in Lemma 2.8.2.

This provides another useful perspective on the functor category Set^K. Indeed, one can reformulate the Kripke semantics for simple type theory entirely in terms of discrete opfibrations $\pi : P \to K$ in place of (covariant) presheaves $P : K \to \mathsf{Set}$. This will be particularly useful when we consider the semantics of *dependent* type theory in the next chapter.

Corollary 2.8.5. *The categories* $\mathsf{dopFib}/_K$ *are sufficient for* λ*-theories: if an equation* $\Gamma \mid s = t : A$ *is not provable in* $\mathbb{T}$*, then there is a* $\mathbb{T}$*-model in discrete opfibrations over a poset* K *in which the equation fails.*

The analogous statement regarding inhabitation of course also holds. The "fibrational" point of view is pursued in [AR11], which also includes details of the dependently typed case.

Exercise 2.8.6. Show that the category Set is *not* sufficient for arbitrary λ-theories, with basic types, terms and equations. Is Pos? (*Hint:* Give a Kripke counter-model of triviality for reflexive domains.)

Sheaves

The category Pos of posets may be cartesian closed, but its slices $\mathsf{Pos}/_P$ are in general not so (by an argument similar to the one given in [Pal03]). By contrast, the (wide but not full) subcategory dopFib of posets and discrete opfibrations is *not* cartesian closed (proof!), whereas its slices $\mathsf{dopFib}/_K \simeq \mathsf{Set}^K$ always are so. Something similar happens with topological spaces: the category Top of all spaces and continuous maps is itself not even cartesian closed (unlike Pos), nor are its slices, but there is a (wide but not full) subcategory $\mathsf{LocHom} \hookrightarrow \mathsf{Top}$ the *slices* of which are always cartesian closed, even though the total category LocHom itself is not. As in the case of dopFib, this is most easily seen by showing that each slice $\mathsf{LocHom}/_X$ is actually equivalent to a functor category known to be cartesian closed, namely the category $\mathsf{Sh}(X)$ of *sheaves* on the space X.

A little topos theory: sheaves and local homeomorphisms.

Definition 2.8.7. A *sheaf* on a space X is a presheaf $F : \mathcal{O}(X)^{\mathsf{op}} \to \mathsf{Set}$ that satisfies the following "patching" condition: for every open cover $U = \bigcup_{i\in I} U_i$ the canonical map

$$FU \to \textstyle\prod_i FU_i \rightrightarrows \prod_{i,j} F(U_i \cap U_j)$$

is an equalizer.

In words, given a family of elements $f_i : \mathsf{y}U_i \to F$ that "match" on the overlaps,

$$f_i|_{U_j} = f_j|_{U_i} : \mathsf{y}(U_i \cap U_j) \to F,$$

there is a unique "amalgamation" $f : \mathsf{y}U \to F$ that restricts to the given family on the cover, $f|_{U_i} = f_i : \mathsf{y}U_i \to F$. For example, in the case of two open sets U, V, this condition says exactly that the following *pushout* diagram in $\mathcal{O}(X)$

$$\begin{array}{ccc} U \cap V & \longrightarrow & V \\ \downarrow & & \downarrow \\ U & \longrightarrow & U \cup V \end{array} \tag{2.10}$$

is taken by F to a *pullback* in Set:

$$\begin{array}{ccc} F(U \cap V) & \longleftarrow & FV \\ \uparrow & & \uparrow \\ FU & \longleftarrow & F(U \cup V) \end{array} \tag{2.11}$$

A basic example of a sheaf is the presheaf $\mathsf{Top}(-, Z) : \mathcal{O}(X)^{\mathsf{op}} \to \mathsf{Set}$ of continuous functions into a fixed space Z, where the open sets $U \subseteq X$ are regarded as subspaces and therefore objects in Top. Indeed, given a family of continuous functions $f_i : U_i \to Z$ that match on the intersections $U_i \cap U_j$, we can define an amalgamation $f : U \to Z$ by setting $f(x) = f_i(x)$ for *some* i with $x \in U_i$ (which exists since $U = \bigcup_{i \in I} U_i$) and the specification will be unique by the matching condition.

It is not difficult to prove the following fact directly from the elementary definition 2.8.7.

Proposition 2.8.8. *The full subcategory* $\mathsf{Sh}(X) \hookrightarrow \mathsf{Set}^{\mathcal{O}(X)^{\mathsf{op}}}$ *is cartesian closed, with the structure inherited from presheaves.*

Exercise 2.8.9. Prove this by showing that Z^Y is a sheaf as soon as Z is a sheaf, by analyzing the exponential as $Z^Y(U) \cong \mathsf{Hom}(\mathsf{y}U \times Y, Z)$.

There is an equivalent perspective on sheaves over X that is often useful, namely as certain spaces over X via certain "generalized covering spaces" $p : Y \to X$ called local homeomorphisms.

Definition 2.8.10. A continuous function $p : Y \to X$ is a *local homeomorphism* if for every $y \in Y$ there is an open set $y \in U \subseteq Y$ such that (i) the image $p(U) \subseteq X$ is open, and (ii) the restriction $p|_U : U \to p(U)$ is a homeomorphism.

Let $\mathsf{LocHom} \hookrightarrow \mathsf{Top}$ be the subcategory of spaces and local homeomorphisms, and $\mathsf{LocHom}/_X$ the slice category. Thus a map $f : (Y, p) \to (Z, q)$ of local homeomorphisms over X is a commutative triangle in Top with $p : Y \to X$ and $q : Z \to X$ local homeomorphisms.

$$\begin{array}{ccc} Y & \xrightarrow{\quad f \quad} & Z \\ & {\scriptstyle p} \searrow \quad \swarrow {\scriptstyle q} & \\ & X & \end{array}$$

One can show that in fact every merely *continuous* map $f : Y \to Z$ making a commutative triangle $q \circ f = p$ is also a local homeomorphism (exercise!), so that this definition is indeed the slice category $\mathsf{LocHom}/_X$, the inclusion of which is *full* in $\mathsf{Top}/_X$.

Given any space "over X" via a continuous map $p_Y : Y \to X$, we can define the *presheaf of local sections* $\Gamma(Y)$ by

$$\Gamma(Y)(U) = \mathsf{Top}/_X(U, Y) \qquad \text{for } U \hookrightarrow X\,.$$

That is, we regard the open set $U \subseteq X$ as a space over X via its inclusion $U \hookrightarrow X$, and consider all "local sections of Y over U", i.e. continuous maps over X from $U \hookrightarrow X$ to $p_Y : Y \to X$. The presheaf $\Gamma(Y)$ is now easily seen to be a sheaf (much like the example $\mathsf{Top}(-, Z)$ above). In this way we have a functor

$$\Gamma : \mathsf{Top}/_X \longrightarrow \mathsf{Set}^{\mathcal{O}(X)^{\mathsf{op}}}$$

which in fact factors through the full subcategory of sheaves $\mathsf{Sh}(X) \hookrightarrow \mathsf{Set}^{\mathcal{O}(X)^{\mathsf{op}}}$. There is also a functor coming back

$$\Lambda : \mathsf{Set}^{\mathcal{O}(X)^{\mathsf{op}}} \longrightarrow \mathsf{Top}/_X,$$

(called the *bundle of germs*), which factors through the full subcategory of local homeomorphisms $\mathsf{LocHom}/_X \hookrightarrow \mathsf{Top}/_X$. The local homeomorphism $\Lambda(P) \to X$ has the total space

$$\Lambda(P) = \coprod_{x \in X} \mathsf{germ}_x(P),$$

where the "stalk of germs at x" $\mathsf{germ}_x(P)$ is defined by

$$\mathsf{germ}_x(P) = \operatorname*{colim}_{U \ni x} PU\,,$$

the colimit being taken over the filter of all open sets U with $x \in U$. The space $\Lambda(P)$ is topologized by declaring as basic opens the images of all partial sections $s : U \to \Lambda(P)$ over X (for $U \subseteq X$ open).

The situation is summarized in the following proposition, for a detailed proof of which, see [MM92, Ch. II].

Proposition 2.8.11. *The bundle of germs functor Λ, which takes a presheaf P on the space X to the local homeomorphism $\Lambda(P) \to X$, is left adjoint to the presheaf of sections functor Γ. The images of these functors are the full subcategories of sheaves $\mathsf{Sh}(X)$, for Γ, and local homeomorphisms $\mathsf{LocHom}/_X$, for Λ. The inclusions have adjoints: a left adjoint $\mathsf{a} \dashv i$ for $\mathsf{Sh}(X)$ and a right adjoint $j \dashv \mathsf{b}$ for $\mathsf{LocHom}/_X$.*

$$\begin{array}{ccc}
\mathsf{Set}^{\mathcal{O}(X)^{\mathsf{op}}} & \underset{\Lambda}{\overset{\Gamma}{\leftarrow\!\!\rightarrow}} & \mathsf{Top}/_X \\
\mathsf{a} \downarrow \quad \uparrow i & & j \uparrow \quad \downarrow \mathsf{b} \\
\mathsf{Sh}(X) & \underset{\Lambda}{\overset{\Gamma}{\leftarrow\!\!\sim\!\!\rightarrow}} & \mathsf{LocHom}/_X
\end{array} \tag{2.12}$$

Thus there is an equivalence of categories $\mathsf{Sh}(X) \simeq \mathsf{LocHom}/_X$.

One immediate consequence is that every slice $\mathsf{LocHom}/_X$ is cartesian closed, by Proposition 2.8.8. Another application is an explicit description of the left adjoint *sheafification* functor $\mathsf{a} = \Gamma \circ \Lambda$, which is seen to preserve finite limits, since each of its factors does so. Another application that will be of use in the semantics of the λ-calculus with sums in the next section is the description of the coproduct of two local homeomorphisms $A \to X$ and $B \to X$ as $A + B \to X$, constructed in $\mathsf{Top}/_X$ in the obvious way.

Corollary 2.8.12. *The sheafified Yoneda embedding* $\mathsf{ay} : \mathcal{O}(X) \to \mathsf{Sh}(X)$ *is still full and faithful and still preserves all limits and exponentials. It now also preserves joins,*

$$\mathsf{ay}U \cup \mathsf{ay}V \cong \mathsf{ay}(U \cup V) \quad \textit{in } \mathsf{Sub}_{\mathsf{Sh}(X)}(1)\,,$$

and similarly for all joins $U = \bigcup_{i\in I} U_i$ *in* $\mathcal{O}(X)$*. Indeed, the factorization*

$$\mathsf{ay} : \mathcal{O}(X) \to \mathsf{Sub}_{\mathsf{Sh}(X)}(1)$$

is an isomorphism of complete Heyting algebras.

Proof. To give a sketch: chasing around the diagram (2.12), we can regard the sheafified Yoneda embedding ay as being given by $\Lambda \circ \mathsf{y} : \mathcal{O}(X) \to \mathsf{LocHom}/_X$, which is just the functor

$$(U \subseteq X) \longmapsto (U \hookrightarrow X)\,,$$

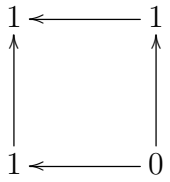
taking an open subset U of X to the inclusion of the open subspace $U \hookrightarrow X$, which is obviously a local homeomorphism. The join of a family $U_i \hookrightarrow X$ of subobjects of 1 in $\mathsf{LocHom}/_X$ is computed there by first taking the coproduct in $\mathsf{Top}/_X$ to give a local homoeomorphism $\coprod_i U_i \to X$ with a disjoint sum of spaces as its domain, and then the image factorization $\coprod_i U_i \twoheadrightarrow \bigcup_i U_i \rightarrowtail X$ to give the open set inclusion $\bigcup_i U_i \hookrightarrow X$, which is the inclusion of the join of the U_i in $\mathcal{O}(X)$. □

Exercise 2.8.13. Show that every representable functor $\mathsf{y}(U)$ is a sheaf. Conclude that the "sheafified" Yoneda embedding $\mathsf{a} \circ \mathsf{y} : \mathcal{O}X \to \mathsf{Sh}(X)$ is fully faithful and injective on objects.

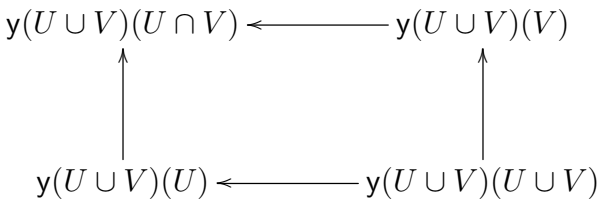
Let us consider a simple example of the preservation of joins by comparing the presheaf join $\mathsf{y}U \cup \mathsf{y}V$ with the sheaf $\mathsf{y}(U \cup V)$, in the case where neither $U \subseteq V$ nor $V \subseteq U$. We can evaluate the presheaf $\mathsf{y}U \cup \mathsf{y}V$ at the pushout diagram (2.10) to get the following diagram of sets,

$$\begin{array}{ccc} (\mathsf{y}U \cup \mathsf{y}V)(U \cap V) & \longleftarrow & (\mathsf{y}U \cup \mathsf{y}V)(V) \\ \uparrow & & \uparrow \\ (\mathsf{y}U \cup \mathsf{y}V)(U) & \longleftarrow & (\mathsf{y}U \cup \mathsf{y}V)(U \cup V) \end{array}$$

which evaluates to the following,



since $(U \cup V)$ is "in" neither $\mathsf{y}U$ nor $\mathsf{y}V$, and so not in their join $\mathsf{y}U \cup \mathsf{y}V$ (where being "in the presheaf" means that the presheaf is not empty at that argument). This diagram is clearly not a pullback, as required for $\mathsf{y}U \cup \mathsf{y}V$ to be a sheaf. On the other hand, evaluating $\mathsf{y}(U \cup V)$ instead results in the evident pullback:



which has $1 = \{*\}$ at all four corners.

Remark 2.8.14. An application of Corollary 2.8.12 is a completeness theorem for full intuitionistic *propositional* logic with respect to categories of sheaves $\mathsf{Sh}(X)$, using the sheafified Yoneda embedding in place of Joyal's representation theorem, which we used for completeness with respect to *pre*sheaves. We leave it to the reader to fill in the details.

Exercise 2.8.15. Fill in the details. (*Hint*: The downset embedding of a Heyting algebra H into the complete Heyting algebra of its ideals $\mathsf{Idl}(H)$ is an injective Heyting homomorphism, and there is an isomorphism of complete Heyting algebras $\mathsf{Idl}(H) \cong \mathcal{O}\mathsf{Spec}(H)$, where the space $\mathsf{Spec}(H)$ is the *prime spectrum* of H, a generalization of the Stone space of a Boolean algebra.)

As a further corollary of Proposition 2.8.11 we have a completeness theorem analogous to Corollary 2.8.5 for models of the λ-calculus in categories of the form $\mathsf{Sh}(X) \simeq \mathsf{LocHom}/_X$, by first deriving completeness for categories of sheaves $\mathsf{Sh}(X)$ from the same covering theorem 2.7.3 that was used for Kripke completeness of the λ-calculus; see Remark 2.7.5. (The completeness theorem with respect to *spaces* rather than posets is proved using a refinement of the Joyal-Tierney covering theorem 2.7.3 due to Moerdijk [**?**], also see [Awo00, **?**].) We will state this specialization where it is needed below for the semantics of λ-calculus with sums. While it is of interest to know that semantics in spaces and local homeomorphisms is sufficient for theories in the λ-calculus, it is also of practical use simply to know that the λ-calculus can be used as an *internal language* to reason about sheaves over a space—a setting with many applications in everyday mathematics.

Equilogical spaces

See [**?**].

2.9 Extensions of the λ-calculus

We conclude our study of simple type theory by considering a few extensions of the basic λ-calculus:

1. λ-Calculus with sums.
2. Natural numbers objects.
3. Higher-order logic.
4. Modal operators.

The first three of these are discussed in much more detail in the book [LS88]. These extensions are not λ-theories as previously defined, but instead involve further operations on types, which may be regarded as (proof-relevant versions of) general rules of inference. Their categorical semantics require additional structures on a CCC.

2.9.1 λ-calculus with sums

We can extend the Curry-Howard correspondence between positive propositional calculi and CCCs by adding "sums" to the CCCs to obtain a categorified version of Heyting algebras, or intuitionistic propositional calculi, which we shall call *BiCartesian Closed Categories* (BiCCCs),

$$\frac{\mathsf{BiCCC}}{\mathsf{CCC}} = \frac{\mathsf{IPC}}{\mathsf{PPC}}.$$

The internal language of such categories will be a simple type theory given by adding "stable sums" 0 and $A + B$ to the λ-calculus. Following [LS88, ADHS01, FDCB02], the additional rules required are the following.

1. The *types* are extended by adding the type constructors $\mathtt{0}$ and $A + B$, so we now have:

   $$\text{Simple types} \quad A ::= \mathtt{B} \mid \mathtt{1} \mid A_1 \times A_2 \mid A_1 \to A_2 \mid \mathtt{0} \mid A_1 + A_2$$

   with the expected formation rules.

2. For the *terms* we now have

   $$\text{Terms} \quad t ::= v \mid c \mid * \mid \langle t_1, t_2 \rangle \mid \mathtt{fst}\, t \mid \mathtt{snd}\, t \mid t_1\, t_2 \mid \lambda x : A\,.\, t \mid !\, t \mid \mathtt{inl}\, t \mid \mathtt{inr}\, t \mid [x.t_1, x.t_2]u$$

   The copairs $[x.t_1, x.t_2]u$ are sometimes called "cases", and the variable x is bound. Their typing rules, and those for the injections $!\, t$, $\mathtt{inl}\, t$, and $\mathtt{inr}\, t$, are:

   $$\frac{\Gamma \mid u : \mathtt{0}}{\Gamma \mid !\, u : C} \qquad \frac{\Gamma \mid a : A}{\Gamma \mid \mathtt{inl}\, a : A + B} \qquad \frac{\Gamma \mid b : B}{\Gamma \mid \mathtt{inr}\, b : A + B}$$

   $$\frac{\Gamma, x : A \mid s : C \qquad \Gamma, y : B \mid t : C \qquad \Gamma \mid u : A + B}{\Gamma \mid [x.s, y.t]u : C}$$

3. The *equations* for these terms are as follows.

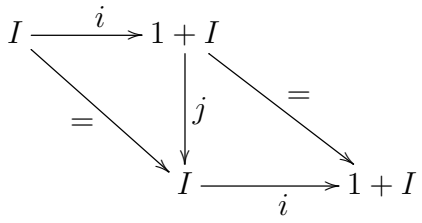
$$\overline{z : C \mid z = !\, u : C} \qquad \overline{[x.a, y.b](\mathtt{inl}\, s) = a[s/x] : C} \qquad \overline{[x.a, y.b](\mathtt{inr}\, t) = b[t/y] : C}$$

$$\overline{u = [x.\mathtt{inl}\, x, y.\mathtt{inr}\, y]u : A + B}$$

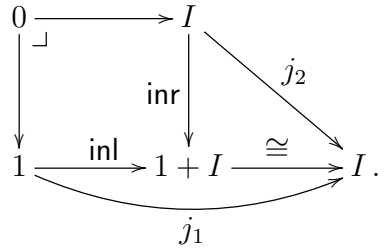
$$\overline{v\big([x.s, y.t]u\big) = [x.vs, y.vt]u : D}$$

The last equation is a "distributivity" law, in which $v : C \to D$ and $x, y \notin \mathsf{FV}(v)$.

4. For an example of a *theory* in the λ-calculus with sums, consider the notion of an *infinite* object I. According to R. Dedekind [Ded88], an object is *infinite* if it admits an injective mapping to a proper subobject. This condition can be captured in a BiCCC by requiring an isomorphism $1 + I \cong I$. Specifically, we require maps and equations as follows:



For then $j = [j \circ \mathsf{inl}, j \circ \mathsf{inr}] : 1 + I \to I$ for unique maps $j_1 = j \circ \mathsf{inl} : 1 \to I$ and $j_2 = j \circ \mathsf{inr} : I \to I$, whence $j_2 : I \to I$ is injective (as a composite of injections), and there is at least one element $j_1 : 1 \to I$ that is not in its image, by the disjointness of coproducts,



Exercise 2.9.1. Write down the theory of infinite objects in the λ-calculus with sums and prove that every model in a BiCCC is indeed Dedekind infinite. Also formulate the theory of a "successor algebra" as an object X equipped with a point $x : 1 \to X$ and an endomorphism $s : X \to X$. Prove that the natural numbers are initial among all successor algebras in Set (with the evident definition of algebraic homomorphisms). Show from this that the natural numbers are Dedekind infinite.

We now have the expected extension of the foregoing results for λ-calculi and CCCs to the case of λ-calculi with sums and BiCCCs, namely:

Proposition 2.9.2. *For any theory $\mathbb{T}$ in λ-calculi with sums, there is a (syntactic) BiCCC $\mathcal{B}_\mathbb{T}$ that classifies $\mathbb{T}$-models in arbitrary BiCCCs $\mathcal{B}$,*

$$\mathsf{BiCCC}(\mathcal{B}_\mathbb{T}, \mathbb{B}) \simeq \mathsf{Mod}(\mathbb{T}, \mathcal{B})\,.$$

In particular, the λ-calculus with sums is complete (in the sense of Corollary 2.4.7) with respect to models in bicartesian closed categories.

The proof is analogous to the previous case, although some care is required with the initial object, the disjointness of sums, and their stability under products with a fixed object (see [LS88, FDCB02]). There is also an internal language correspondence as in Section 2.5 that we need not spell out. An interesting question considered in [FDCB02] is that of *type isomorphisms*, as a generalization of elementary algebraic equations. For example ...

Remark 2.9.3 (Variable set completeness)**.** Completeness of λ-calculus with respect to arbitrary presheaf categories $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ is apparently more difficult to generalize to λ-calculus with sums than was the general categorical completeness theorem, Proposition 2.9.2. This is because, although such categories $\mathsf{Set}^{\mathbb{C}^{\mathsf{op}}}$ have very well-behaved (indeed, freely added) coproducts, the Yoneda embedding does not preserve the coproducts that may exist in the index category $\mathbb{C}$. In the "proof-irrelevant" propositional case, we solved this problem using a more sophisticated embedding theorem due to Joyal, Theorem **??**. One may conjecture that something similar could hold in the present case: we can embed a BiCCC $\mathcal{B}$ into the category of presheaves on all "biCartesian" functors $M : \mathcal{B} \to \mathsf{Set}$ (playing the role of the prime filters in a Heyting algebra); however, we would still need to show that this analogous "evaluation embedding" also preserves all exponentials; whether this holds is an open question.

We leave the question of completeness with respect to variable sets aside for now, and briefly consider a different approach to the semantics of BiCCCs using *sheaves*, for which one can show completeness using well-known results (e.g. [FS99]).

Definition 2.9.4. Let $\mathcal{B}$ be a BiCCC. A presheaf $F : \mathcal{B}^{\mathsf{op}} \to \mathsf{Set}$ is called a *sheaf* (for the $+$-topology) if it preserves finite products. Explicitly, a sheaf F is a contravariant functor that takes the finite coproducts in $\mathcal{B}$ to products in Set (via the canonical maps),

$$\begin{aligned} F(0) &\cong 1\,, \\ F(A+B) &\cong FA \times FB\,. \end{aligned}$$

A *morphism of sheaves* $f : G \to F$ is just a natural transformation.

Lemma 2.9.5. *By definition, the category of sheaves is a full subcategory. The inclusion $i : \mathsf{Sh}(\mathcal{B}) \hookrightarrow \widehat{\mathcal{B}}$ has a left adjoint,*

$$\mathsf{a} : \widehat{\mathcal{B}} \longrightarrow \mathsf{Sh}(\mathcal{B}),$$

called sheafification*, which, moreover, preserves all finite limits.*

Proof. Since finite products commute with limits, it is easy to see that the FP-functors are closed under all limits. So by the adjoint functor theorem, we see that the full subcategory of FP-functors is reflective in the category of all Set-valued functors, as we have already shown in Section **??**. That the reflector a preserves finite limits can be shown by analyzing the sheafification functor a in terms of the so-called Grothedieck +-construction; see [MM92, III.5]. □

Proposition 2.9.6. *We require the following facts about the subcategory category*

$$\mathsf{Sh}(\mathcal{B}) \hookrightarrow \widehat{\mathcal{B}}$$

of +-sheaves on a BiCCC $\mathcal{B}$.

1. *The representable functors $\mathsf{y}B : \mathcal{B}^{\mathsf{op}} \to \mathsf{Set}$ are all sheaves; in particular, the terminal object $1 = \mathsf{y}1$ is a sheaf.*

2. *The sheafified Yoneda embedding $\mathsf{ay} : \mathcal{B} \to \mathsf{Sh}(\mathcal{B})$ is fully faithful, and preserves finite coproducts.*

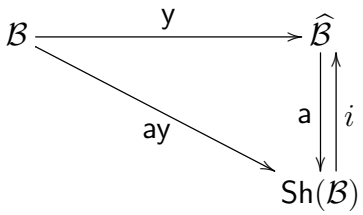
3. *If F, G are sheaves, so is their product $F \times G$, and if G is a sheaf and F a presheaf, then the presheaf exponential G^F is a sheaf.*

Thus in particular, the fully faithful functor $\mathsf{ay} : \mathcal{B} \hookrightarrow \mathsf{Sh}(\mathcal{B})$ preserves the BiCCC structure.

Proof. 1. For any object $B \in \mathcal{B}$ we have:

$$\begin{aligned} \mathsf{y}B(0) &= \hom(0, B) = 1\,, \\ \mathsf{y}B(A_1 + A_2) &= \hom(A_1 + A_2, B) \cong \hom(A_1, B) \times \hom(A_2, B) \\ &= \mathsf{y}B(A_1) \times \mathsf{y}B(A_2)\,. \end{aligned}$$

2. To see that $\mathsf{ay} : \mathcal{B} \to \mathsf{Sh}(\mathcal{B})$ is fully faithful, note that by (1) we have $\mathsf{y} \cong i \circ (\mathsf{ay}) : \mathcal{B} \to \widehat{\mathcal{B}}$, which is fully faithful, and $i : \mathsf{Sh}(\mathcal{B}) \to \widehat{\mathcal{B}}$ is so as well.



To see that ay preserves sums, for any sheaf F, we have:

$$\begin{aligned} \mathsf{Sh}(\mathcal{B})(\mathsf{ay}(0), F) &\cong \widehat{\mathcal{B}}(\mathsf{y}(0), iF) \cong iF(0) \cong 1\,, \\ \mathsf{Sh}(\mathcal{B})(\mathsf{ay}(A + B), F) &\cong \widehat{\mathcal{B}}(\mathsf{y}(A + B), iF) \cong iF(A + B) \cong iF(A) \times iF(B)\,. \end{aligned}$$

3. If F, G are sheaves then $F \times G$ is one as well, since, as presheaves,

$$\begin{aligned}(F \times G)(0) &\cong F0 \times G0 \cong 1 \times 1 \cong 1\,,\\ (F \times G)(A+B) &\cong F(A+B) \times G(A+B) \cong (FA \times FB) \times (GA \times GB)\\ &\cong (F \times G)(A) \times (F \times G)(B)\,.\end{aligned}$$

If G is a sheaf and F a presheaf, then as presheaves,

$$\begin{aligned}(iG^F)(0) &\cong \widehat{\mathcal{B}}\big(\mathsf{y}(0) \times F, iG\big) \cong \mathsf{Sh}(\mathcal{B})\big(\mathsf{a}(\mathsf{y}(0) \times F), G\big)\\ &\cong \mathsf{Sh}(\mathcal{B})\big(\mathsf{ay}(0) \times \mathsf{a}F, G\big) \cong \mathsf{Sh}(\mathcal{B})\big(\mathsf{a}0, G\big) \cong 1\,.\\ (iG^F)(A+B) &\cong \widehat{\mathcal{B}}\big(\mathsf{y}(A+B) \times F, iG\big) \cong \widehat{\mathcal{B}}\big(\mathsf{ay}(A+B) \times F, G\big)\\ &\cong \widehat{\mathcal{B}}\big((\mathsf{ay}(A) + \mathsf{ay}(B)) \times F, G\big)\\ &\cong \widehat{\mathcal{B}}\big((\mathsf{ay}(A) \times F) + (\mathsf{ay}(B) \times F), G\big)\\ &\cong \widehat{\mathcal{B}}\big(\mathsf{ay}(A) \times F, G\big) \times \widehat{\mathcal{B}}\big(\mathsf{ay}(B) \times F, G\big)\\ &\cong \widehat{\mathcal{B}}\big(\mathsf{y}(A) \times F, iG\big) \times \widehat{\mathcal{B}}\big(\mathsf{y}(B) \times F, iG\big)\\ &\cong iG^F(A) \times iG^F(B)\,.\end{aligned}$$

□

As a result of the forgoing embedding theorem, the completeness of the basic λ-calculus with respect to presheaves can be extended to completeness of the λ-calculus *with sums* with respect to categories of *sheaves* (although we have not yet defined these except in the special cases of topological spaces and $+$ sheaves on a category with stable, finite coproducts). To that end, we define a *model* in a category $\mathsf{Sh}(\mathbb{C}, +)$ of sheaves (for the $+$ topology) on a small category $\mathbb{C}$ with stable finite coproducts to be a BiCCC functor from $\mathcal{B}_{\mathbb{T}}$, the classifying BiCCC of a theory $\mathbb{T}$, into $\mathsf{Sh}(\mathbb{C}, +)$. We then have the following completeness theorem.

Proposition 2.9.7 (Completeness of λ-Calculus with Sums)**.** *For a theory $\mathbb{T}$ in the λ-calculus with sums,*

1. *a type A is inhabited in every model in a category of sheaves $\mathsf{Sh}(\mathbb{C}, +)$ iff there is a closed term $a : A$,*

2. *an equation $\Gamma \mid s = t : A$ holds in every model in a category of sheaves $\mathsf{Sh}(\mathbb{C}, +)$ iff there is a proof of it from the equations of $\mathbb{T}$.*

There is a more general notion of a sheaf for a "Grothendieck topology" on a small category $\mathbb{C}$, of which the $+$-topology on a category with sums is a special case, and the foregoing proposition then generalizes to that case, but we shall not pursue this further here.

Exercise 2.9.8. Fill in the details of the proof of Proposition 2.9.7.

Finally, we can again apply the Joyal-Tierney covering theorem 2.7.3 to obtain completeness of the λ-calculus *with sums* with respect to categories $\mathsf{Sh}(X)$ of sheaves on a *space*:

Corollary 2.9.9 (Topological Completeness of λ-Calculus with Sums)**.** *For a theory $\mathbb{T}$ in the λ-calculus with sums,*

1. *a type A is inhabited in every model in a category of sheaves on a space $\mathsf{Sh}(X)$ iff there is a closed term $a : A$,*

2. *an equation $\Gamma \mid s = t : A$ holds in every model in a category of sheaves on a space $\mathsf{Sh}(X)$ iff there is a proof of it from the equations of $\mathbb{T}$.*

Of course, the result can also be formulated equivalently in terms of BiCCCs of the form $\mathsf{LocHom}/_X$ rather than $\mathsf{Sh}(X)$, where the coproducts of local homeomorphisms $A \to X$ and $B \to X$ are more easily constructed naively as $A + B \to X$ in the underlying category Top/X (by Proposition 2.8.11).

Remark 2.9.10. It may also be asked whether there is a *single space* X such that $\mathsf{Sh}(X)$ is sufficient for all theories in the λ-calculus with sums, as has been shown e.g. for intuitionistic first-order logic IFOL with respect to just sheaves on the real line $\mathbb{R}$ [].

2.9.2 Natural numbers objects

Using sums we can describe infinite objects $X + 1 \cong X$, but we cannot describe the *free* such objects, such as the *initial* one, without having a general induction principle with respect to other such objects. Such inductive type are more conveniently formulated in dependent type theory, as we shall do in the next chapter, but we can also formulate them in simple type theory by adding new recursion operations, see Lambek-Scott [LS88]. . This leads to the important notion of a *natural numbers object*: an initial infinite object.

2.9.3 Higher-order logic

This example presumes familiarity with the results of Chapter **??**, or at least with the basic categorical approach to first-order logic as presented in [MM92, **?**].

The approach to IHOL presented here is closely tied to *topos theory*, which is to be treated in greater depth in Chapter **??**. Also see Lambek-Scott [LS88].

Remark 2.9.11.

2.9.4 Modalities

Recall first the propositional modal logics IS4, IS5 with adjoints, natural deduction using Bierman-dePaiva, Kavvos.

Example of a CCC with a "modal operator": pointed sets for partial functions and the lifting monad.

Summarize Moggi's modal λ-calculus. Also see Shulman.

See [Sco80b, Sco80a] for more on the λ-calculus

Still ToDo: Normalization

Bibliography

[ADHS01] T. Altenkirch, P. Dybjer, M. Hofmann, and P. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science (LICS 2001)*, pages 303–310, 2001.

[AGH24] S. Awodey, N. Gambino, and S. Hazratpour. Kripke-Joyal forcing for type theory and uniform fibrations, 2024. Preprint available as `https://arxiv.org/abs/2110.14576`.

[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.

[Baua] A. Bauer. On a proof of cantor's theorem. Blogpost at `https://math.andrej.com/2007/04/08/on-a-proof-of-cantors-theorem`.

[Baub] A. Bauer. On fixed-point theorems in synthetic computability. Blogpost at `https://math.andrej.com/2019/11/07/on-fixed-point-theorems-in-synthetic-computability`.

[Coq22] Thierry Coquand. Type theory, 2022. The Stanford Encyclopedia of Philosophy, `https://plato.stanford.edu/archives/fall2022/entries/type-theory`.

[Ded88] R. Dedekind. *Was sind und was sollen die Zahlen?* Vieweg, 1888.

[FDCB02] M. Fiore, R. Di Cosmo, and V. Balat. Remarks on isomorphisms in typed lambda calculi with empty and sum types. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science (LICS 2002)*, pages 147–156, 2002.

[Fri75] H. Friedman. Equality between functionals. In R. Parikh, editor, *Logic Colloquium*. Springer-Verlag, New York, 1975.

[FS99] Marcelo Fiore and Alex Simpson. Lambda definability with sums via Grothendieck logical relations. In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications*, pages 147–161, Berlin, Heidelberg, 1999. Springer.

[Joh03] P.T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium, 2 vol.s*. Number 43 in Oxford Logic Guides. Oxford University Press, 2003.

[JT84] A. Joyal and M. Tierney. *An extension of the Galois theory of Grothendieck*. Memoirs of the AMS. American Mathematical Society, 1984.

[Law69] F.W. Lawvere. Diagonal arguments and cartesian closed categories. In *Category Theory, Homology Theory and their Applications II*, volume 92 of *Lecture Notes in Mathematics*. Springer, Berlin, 1969. Reprinted with author commentary in *Theory and Applications of Categories* (15): 1–13, (2006).

[LS88] J. Lambek and P.J. Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge, 1988.

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

[ML84] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory*. Bibliopolis, 1984.

[MM92] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic. A First Introduction to Topos Theory*. Springer-Verlag, New York, 1992.

[Pal03] Erik Palmgren. Groupoids and local cartesian closure. 08 2003. unpublished.

[Plo73] G. D. Plotkin. Lambda-definability in the full type hierarchy. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, New York, 1973.

[Sco70] Dana S. Scott. Constructive validity. In M. Laudet, D. Lacombe, L. Nolin, and M. Schützenberger, editors, *Symposium on Automatic Demonstration*, volume 125, pages 237–275. Springer-Verlag, 1970.

[Sco80a] Dana S. Scott. The lambda calculus: Some models, some philosophy. In *The Kleene Symposium*, pages 223–265. North-Holland, 1980.

[Sco80b] Dana S. Scott. Relating theories of the lambda calculus. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 403–450. Academic Press, 1980.

[Sim95] A. Simpson. Categorical completeness results for the simply-typed lambda-calculus. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, pages 414–427. Springer, 1995.

[Tai68] William W. Tait. Constructive reasoning. In *Logic, Methodology and Philos. Sci. III (Proc. Third Internat. Congr., Amsterdam, 1967)*, pages 185–199. North-Holland, Amsterdam, 1968.