

# Fibration Representations of the Lambda Calculus

---

A Thesis  
Presented to  
The Division of Mathematics and Natural Sciences  
Reed College

---

In Partial Fulfillment  
of the Requirements for the Degree  
Bachelor of Arts

---

Michael J. Lieberman

May 2003



Approved for the Division  
(Mathematics)

---

Thomas W. Wieting



# Acknowledgments

A number of people have been indispensable to the completion of this thesis in its present form. I am indebted to David Perkinson for encouraging me in my desire to pursue a thesis in the field of categorical logic and for putting me in touch with Steven Awodey at Carnegie Mellon. He in turn merits many thanks, both for his suggestion of the topic for this thesis, and for guiding me from afar over the course of the year. Above all, though, I would like to thank my advisor, Tom Wieting, who has provided me with a good deal of useful advice and exhibited a seemingly limitless tolerance for my yearlong flight into abstraction.

Among those who have contributed indirectly to this document are my good friends and fellow students, particularly Godfrey Leung, Daniel Spoth, and Neelaksh Sadhoo, who have provided much-needed and much-appreciated distractions from it. I am also grateful to my family, who instilled in me my love of knowledge and steered me onto the path that I now tread. In particular, of course, I owe a debt of gratitude to my parents, Jay and Sally, and my sister, Hannah, without whose support and understanding none of this would have been possible.



# Table of Contents

<b>Introduction</b> . . . . .	<b>1</b>
<b>1 The Lambda Calculus</b> . . . . .	<b>5</b>
1.1 The Untyped Lambda Calculus . . . . .	6
1.2 The Simply-typed Calculus . . . . .	12
1.3 Untyped as Typed . . . . .	17
<b>2 Elementary Categorical Semantics</b> . . . . .	<b>21</b>
2.1 Lambda Theories, Cartesian Closed Categories . . . . .	25
2.2 Models In Arbitrary CCCs . . . . .	30
2.3 Presheaf Semantics . . . . .	34
<b>3 The Sheaf Representation</b> . . . . .	<b>39</b>
3.1 Sheaves . . . . .	39
3.2 The Representation Theorem . . . . .	43
3.3 The Improved Presheaf Representation . . . . .	49
<b>4 Fibration Semantics</b> . . . . .	<b>51</b>
4.1 Poset Fibrations . . . . .	52
4.2 The Fibration Representation . . . . .	65
<b>Bibliography</b> . . . . .	<b>73</b>



# Abstract

We enquire into the categorical semantics of the simply typed  $\lambda$ -calculus, establishing the strong completeness of fibrational poset semantics and proving, moreover, that every  $\lambda$ -theory has a representation of this form. The program of this thesis falls naturally into four parts. In Chapter 1, the syntax of the simply-typed  $\lambda$ -calculus is introduced by way of the type-free calculus, which is then shown to be a special case. Chapter 2 discusses the notions of semantic completeness that will be employed throughout, and presents a first pass at a workable system of semantics, first in the category of cartesian closed categories, then in the presheaves on such categories. In Chapter 3, the recent sheaf representation theorem due to Steven Awodey is introduced, along with a sketch of its proof. Categories of fibrations over partially ordered sets are examined in Chapter 4, and it is proven that the class of models in such categories constitutes strongly complete semantics for the simply-typed  $\lambda$ -calculus.



# Introduction

In some ways, it seems almost silly to inquire after a formal characterization of an object as seemingly intuitive as a function. After all, the essential notion is familiar to anyone who possesses even a passing acquaintance with mathematics, and is employed without hesitation by mathematicians themselves. Upon closer examination, however, one finds that among the current theories that purport to provide a suitable foundation for mathematics, namely set theory and category theory, there is a certain lack of uniformity in the treatment of functions. Given this situation, a preoccupation with formal theories of functions seems a great deal more reasonable. The simply-typed lambda calculus, which is to be the subject of this thesis, is just such a theory. Indeed, it is an exceptionally pure theory in so far as it is concerned, as it were, with functions in themselves. In the hope of motivating the more abstract and detailed discussion that is to follow, the aim of this brief introduction is to provide a conceptual account of the lambda calculus, in the spirit of [15], developing it in relation to the more familiar theories of functions associated with the set-theoretic and category-theoretic perspectives.

Let us begin by considering the treatment of functions in the more familiar framework of a theory of sets, assuming for the sake of convenience that the theory is of the kind proposed by Zermelo and Fraenkel. In this context, a function between sets  $A$  and  $B$  may be regarded as a rule that assigns precisely one element of  $B$  to each element of  $A$ , with identity functions and compositions defined in the natural way. In addition, the theory is sufficiently rich that, given any pair of sets  $A$  and  $B$ , it encompasses all possible functions between them. One should note, however, that the notion of a function is a subsidiary, dependent one, inasmuch as the principal objects of study are the sets themselves. Indeed, one finds that functions  $f$  between sets  $A$  and  $B$  are identified with their graphs, subsets of the Cartesian product  $A \times B$ . This secondary status may be taken as an essential feature of the set-theoretic interpretation of functions. One should also note that there is a certain

ambiguity in the specification of the codomain associated with a particular function. To take a simple example, the successor function  $\sigma : x \mapsto x + 1$  on  $\mathbf{N}$ , the set of natural numbers, might be regarded as having as its codomain any one of the sets  $\{x \in \mathbf{N} \mid x \geq 1\}$ ,  $\mathbf{N}$ , or  $\mathbf{Z}$ , the set of all integers. At this point, one might proceed to investigate the functions that arise in extended versions of set theory, such as class theory, but for our purposes it is better to turn to a more markedly different perspective.

As Dana Scott has pointed out, an arbitrary category yields a functional theory, provided that we interpret the objects of the category as domains and the morphisms of the category as functions. The theories that arise in this fashion do bear some resemblance to the set-based ones since, by the definition of a category, there is an identity function (morphism) on each domain (object) and the composition of composable functions is itself a function. The distinguishing characteristics of the set-theoretic presentation are not present here, however. Within a given category, one is primarily concerned with the functions, whereas the domains are essentially featureless. Indeed, the domains possess only those properties that are encoded by the functions. Thus we find that the primacy relation between the functions and domains is, in some sense, the reverse of that considered above. In addition, the ambiguity mentioned earlier in reference to the range of a set-theoretic function is not a concern, as each function comes equipped with a fixed domain and codomain, again by the definition of a category.

While one might reasonably say that the resulting theory is a purer one, one must also acknowledge that, in general, it is rather weak. After all, it is perfectly consistent with the category axioms for two domains to have absolutely no functions between them. Nor will such a theory necessarily incorporate functions of higher arity, as there is no guarantee that the category is closed under finite products. Both of these problems disappear, however, if we restrict our attention to the class of cartesian closed categories, or CCCs. These categories will be discussed in greater detail below, but it suffices for the purposes of the present exposition merely to note that if  $\mathbf{C}$  is a cartesian closed category, then for any domains  $A$  and  $B$ , there is a product domain  $A \times B$  and an exponential domain  $B^A$ , where  $B^A$  is interpreted as the collection of functions from  $A$  to  $B$ . With these new constructions, of course, the theory is now rich enough to capture multivariate functions, as well as the notion of an arbitrary function between domains (regarded, roughly speaking, as an element of the appropriate exponential domain). Thus each cartesian closed category gives

rise to a full-fledged theory of functions.

In a way that will be made precise in Chapter 2, theories of the simply-typed lambda calculus are essentially indistinguishable from cartesian closed categories. In particular, we may think of a theory of the lambda calculus as a formal, logical theory that captures the functions, domains, operations and functional equations that arise in a particular cartesian closed category. The language of such a lambda theory consists of a family of terms (which correspond to the functions in the category), each of which is assigned a type (corresponding to a domain). The type assignment is done in the natural way: if, for example,  $f$  is a function between domains  $A$  and  $B$  in the category, the corresponding term  $F$  is of type  $B$  with a single free variable of type  $A$ . Every equation that holds between functions in the category also holds between the relevant terms in the theory. In addition, the principal operations on terms, functional abstraction, application and pairing, correspond to operations on functions in the category. Although the details will become clearer in time, it is useful to keep in mind that any theory of the simply-typed lambda calculus is merely a reformulation of a categorical theory of the kind described above, and thus it shares the same essential features.

Whenever one studies a formal theory, it is natural to ask what its models might look like and, similarly, what the most suitable systems of semantics might be. This question is particularly important in the case of the lambda calculus, since it does not possess any canonical semantics ([1]). Moreover, while we are naturally inclined to look for models in the category of sets and set-maps, this turns out to be misguided in general. From the equivalence between lambda theories and cartesian closed categories, we may infer that every theory has a model in the system of semantics consisting of cartesian closed categories. We would do well to look for other, less trivial semantics, though. In a recent paper, S. Awodey demonstrated that every lambda theory has a model in the category of sheaves over a specially constructed topological space. Since the sheaf construction is such a familiar one, this is a tremendously appealing result. It also leads to new systems of semantics. In the course of this exposition, it will be shown that the completeness results with respect to sheaf semantics can be extended to the more elementary, intuitive semantics consisting of categories of fibrations over partially ordered sets.



# Chapter 1

## The Lambda Calculus

Although it is little known in conventional mathematical circles, the  $\lambda$ -calculus has been a subject of active research since the 1930s. The type-free formulation has its origins in an attempt by Alonzo Church to develop a novel theory that might serve as a foundation for mathematics. Though his theory was ultimately shown to be inconsistent, he was able to salvage a consistent subtheory describing function definition and application. This theory, the first incarnation of the  $\lambda$ -calculus, proved to be remarkably successful. In 1936, for example, Church used it to give a precise characterization of the notion of computability and, as a result, he was able to establish the unsolvability of the Entscheidungsproblem, the problem of determining whether an arbitrary formula in a first order language is provable. Shortly afterward, of course, Turing proved the same result using his better known formalization of computability, defined via the notion of machines. As it turns out, the functions definable within the  $\lambda$ -calculus are precisely the same as those that may be computed by Turing machines, namely the recursive functions. More recently, the type-free  $\lambda$ -calculus has been used in the design and testing of functional programming languages, as it allows a number of theoretical problems to be formulated in a particularly clean fashion.

One could provide innumerable reasons for the introduction of types to Church's calculus, both practical and aesthetic. The clearest reason, perhaps, is that the untyped calculus presents an exceedingly odd view of functions, insofar as any function in the theory may take itself as input. While this peculiar feature proves to be useful in capturing recursion, it runs counter to our deeply ingrained notions about functions. After all, in any familiar mathematical context, a function is a radically different kind of thing than its input or output. If we assign types to the

terms of the calculus, though, and place natural restrictions on the operations, we may produce a theory that is more in line with our intuition. For example, we will restrict the operation of application so that given an arbitrary term  $M$  and a term  $N$  of type  $\sigma$ , the application  $MN$  will be defined only if  $M$  is of a function type  $\sigma \rightarrow \tau$ . Naturally, there are different typing systems. In the  $\lambda$ -calculus with dependent types, for example, the types are not fixed but rather vary with a parameter. In the simply-typed  $\lambda$ -calculus, on the other hand, the types are fixed. In the present account, of course, we will be interested solely in the simply-typed case. Indeed, in all subsequent references to typed  $\lambda$ -calculi, it will be assumed that the types are simple.

The purpose of this chapter is to provide an introduction to the syntax of the typed  $\lambda$ -calculus. The nature of the operations of the  $\lambda$ -calculus is somewhat clearer in the type-free theory, though, and thus it provides a more natural starting point for our inquiry. Ultimately, we will see that the type-free calculus is actually a special case of the typed calculus. While every effort will be made to convey the essential details, the presentation will be far from exhaustive. Readers interested in a more complete treatment of the type-free calculus, or in the myriad applications thereof, may wish to consult [3] or [2]. For additional information on the typed  $\lambda$ -calculus, one might look to [9], [7], or [15].

## 1.1 The Untyped Lambda Calculus

We will begin with a conceptual introduction, before proceeding to a more detailed description of the syntax. Let us start, then, by considering the universe of functions described by the type-free  $\lambda$ -calculus. Every term of the  $\lambda$ -calculus may be regarded as a function, rule, or algorithm that takes a single argument, and is specified completely by the way that it acts on that argument. Moreover, a given term may take any other term as its argument, including itself. The theory is concerned solely with the relations of equivalence between terms and thus it is, in an important sense, a decidedly nonlogical one. Hence any quantifiers and logical connectives that are implicit in the following exposition belong exclusively to the metalanguage in which we discuss the theory.

There are two operations in the  $\lambda$ -calculus: application and lambda abstraction. In the operation of application, a term is supplied with another term to act as its argument. Given any terms  $M$  and  $N$ , when we apply  $M$  to  $N$  we regard  $M$

as the function, the active entity, whereas  $N$  (although it is itself a function) is regarded merely as the input datum, a passive entity. Contrary to our intuition, we should not immediately associate the application of  $M$  to  $N$ , denoted by the simple juxtaposition  $MN$ , with the output of  $M$  on input  $N$ , if only because we do not yet possess any formal means of computing what this output would be. As we will see, this may be accomplished through  $\beta$ -reduction, a notion that will be discussed presently. To simplify the notation in cases where many terms are being applied, it is customary to take the operation of application to be left associative. In other words, given any terms  $M_1, M_2, M_3 \dots, M_n$ , the expression  $M_1 M_2 M_3 \dots M_n$  is understood to mean  $(\dots ((M_1 M_2) M_3) \dots) M_n$ .

In some ways, the operation of lambda abstraction is more interesting than application since, one might say, it is responsible for the distinctive character of the  $\lambda$ -calculus. In essence, lambda abstraction permits us the unrestricted power to create new terms merely by specifying the way in which they act on their arguments. In particular, given any term  $M$ , which may or may not contain a free occurrence of the indeterminate  $x$ , we denote by  $\lambda x.M$  the term that assigns to its argument, say  $N$ , the expression  $M[N/x]$ , which results when all free occurrences of  $x$  in  $M$  are replaced by  $N$  (this will all be made precise in a moment). So, for example, we may define an identity term  $I = \lambda x.x$ , which always returns its argument as output. Roughly speaking, we might also define a term corresponding to the successor function on the positive integers by  $\sigma = \lambda x.(x + 1)$ . This is a tad imprecise, however, since neither the positive integers nor the symbol “+” belong to the language of the  $\lambda$ -calculus, although they may ultimately be defined in terms of longer and more complicated lambda expressions. Since one often has to deal with such expressions involving a number of lambda operators, it is convenient to require that the operation of lambda abstraction be right associative, i.e. for any term  $M$  and indeterminates  $x_1, x_2, \dots, x_n$ , we may abbreviate  $\lambda x_1.(\lambda x_2.(\dots (\lambda x_n.M) \dots))$  by  $\lambda x_1.\lambda x_2 \dots \lambda x_n.M$ , which may be further abbreviated as  $\lambda x_1 x_2 \dots x_n.M$  without giving rise to any ambiguity.

It is natural to think of the  $\lambda x$  as binding the free occurrences of the term on which it operates. In the expression  $x + 1$ , for example,  $x$  is obviously free, whereas it does not make sense to say that  $x$  is free in  $\lambda x.x + 1$ , the function defined by the rule  $x \mapsto x + 1$ . In general, we say that an occurrence of an indeterminate  $x$  in a term  $M$  is bound if it falls within the scope of a  $\lambda x$  operator. Otherwise, we say that the occurrence of  $x$  is free. Hence the variables that occur free in the term  $\lambda x.xyz$

are  $y$  and  $z$ , the only variable that occurs free in  $\lambda yx.xyz$  is  $z$ , and every variable is bound in  $\lambda zy x.xyz$ .

Intuitively, we would like the operations of abstraction and application to work together as in the following relation

$$(\lambda x.M[x])N = M[N/x]$$

for any terms  $M$  and  $N$ , and indeterminate  $x$ , where, once again,  $M[N/x]$  is the expression that results when each free occurrence of  $x$  in  $M$  is replaced by  $N$ . As usual, we must ensure that none of the free variables in  $N$  become bound when this substitution is made, but this can be accomplished by merely renaming the free variables in  $N$  wherever necessary. At any rate, the relation above is the principal axiom scheme of the type-free  $\lambda$ -calculus, commonly referred to as  $(\beta)$ , and gives rise to the notion of  $\beta$ -reduction mentioned above.

Naturally, a theory that only encompasses functions in a single variable will be of little interest, so one might now ask whether the  $\lambda$ -calculus is able to describe functions in multiple arguments, rather than just one. The answer is yes: multivariate functions may be obtained through the process known as “currying,” which consists in the repeated application of lambda operators to the expression corresponding to the function. For example, given an expression in two free variables, say  $f(x, y)$ , the function defined by the assignment  $(x, y) \mapsto f(x, y)$  is the same as the term  $F = \lambda xy.f(x, y)$ . Using the rule  $(\beta)$ , we may verify that this is indeed correct:

$$\begin{aligned} Fxy &= (\lambda x.(\lambda y.f(x, y)))xy \\ &= ((\lambda x.(\lambda y.f(x, y)))x)y \\ &\stackrel{\beta}{=} (\lambda y.f(x, y))y \\ &\stackrel{\beta}{=} f(x, y) \end{aligned}$$

Once one becomes accustomed to the notation, it is clear that this is related to the familiar manoeuvre in multivariable calculus where, given a function  $f$  in the variables  $x$  and  $y$ , we hold  $x$  fixed, thereby obtaining a function that depends solely on  $y$ . When we view  $f$  in this way, it corresponds to the chain of assignments  $x \mapsto (y \mapsto f(x, y))$  or, in lambda notation,  $\lambda xy.f(x, y)$ . Naturally, we may capture functions in more than two variables in precisely the same way, i.e. the function  $(x_1, x_2, \dots, x_n) \mapsto g(x_1, x_2, \dots, x_n)$  corresponds to the term  $G = \lambda x_1 x_2 \dots x_n.g(x_1, x_2, \dots, x_n)$ .

Now that we have a good general understanding of the behavior of the terms and of the operations of application and abstraction, it seems that we should, at long last, obtain a precise syntactic description of the terms and of the equivalence relations that hold between them. We begin by considering the language and, in particular, the symbols from which the terms are constructed.

**Definition 1.1.1 (Alphabet)** *The terms of the  $\lambda$ -calculus are words over the alphabet consisting of a countable family of variables  $X = \{x_1, x_2, \dots\}$ , a countable family of basic terms  $B = \{b_1, b_2, \dots\}$ , and the symbols  $(, )$ , and  $\lambda$ .*

The square brackets  $[$  and  $]$  belong to the metalanguage, and appear only in the context of our discussions of substitution.

Of course, the alphabet is only part of the story. What we require now is a grammar, a set of rules governing the formation of words over the alphabet, which will allow us to recognize and construct the ones that will be meaningful in the context of the theory, namely the terms.

**Definition 1.1.2 (Terms)** *The set of terms, which we will denote by  $\mathcal{L}(T)$ , is constructed inductively according to the following rules:*

1. If  $x \in X$ ,  $x \in \mathcal{L}(T)$
2. If  $b \in B$ ,  $b \in \mathcal{L}(T)$
3. If  $M, N \in \mathcal{L}(T)$ ,  $MN \in \mathcal{L}(T)$
4. If  $x \in X$  and  $M \in \mathcal{L}(T)$ ,  $\lambda x.M \in \mathcal{L}(T)$

Since our discussion has thus far lingered on the general properties of the terms, let us take a moment to consider their concrete appearance. According to the rules above, the following are examples of terms:

$$x \quad b \quad xx \quad xb \quad \lambda x.y \quad \lambda x.xx \quad \lambda x.(\lambda y.(x(yz))) \quad (\lambda z.xz)(\lambda x.(\lambda y.(zx)y))$$

where  $x, y$ , and  $z$  are in  $X$  and  $b$  is in  $B$ . In general, then, the terms are finite strings of constants and variables, which may or may not be bound by lambda operators, or are juxtapositions of such terms. It should be clear now why the expression  $\lambda x.x+1$ , which is not of this form, was introduced with a certain measure of equivocation. When an expression is well-formed, there is nothing to prevent us from expressing it in a more compact way using the abbreviations introduced above. Hence, for

example, the term  $(\lambda z.xz)(\lambda x.(\lambda y.(zx)y))$ , expressed in the canonical fashion, may be rewritten as  $(\lambda z.xz)(\lambda xy.zxy)$ .

Now that we have a precise characterization of the terms of the  $\lambda$ -calculus, it is only fitting that we should revisit our earlier intuitions about free and bound variables. We may now provide a more complete description:

**Definition 1.1.3** *The set of free variables in a term  $M$ , denoted  $FV(M)$ , is defined inductively as follows:*

1.  $FV(x) = \{x\}$

2.  $FV(MN) = FV(M) \cup FV(N)$

- 3.

$$FV(\lambda x.M) = \begin{cases} FV(M) \setminus \{x\} & \text{if } x \in FV(M) \\ FV(M) & \text{otherwise} \end{cases}$$

At first glance, it seems that the term  $(\lambda z.xz)(\lambda x.(\lambda y.zxy))$  contains free occurrences of  $x$  and  $z$ . To see how the definition of the set of free variables works, we may check that  $FV((\lambda z.xz)(\lambda x.(\lambda y.zxy)))$  is, in fact, equal to  $\{x, z\}$ :

$$\begin{aligned} FV((\lambda z.xz)(\lambda x.(\lambda y.zxy))) &= FV((\lambda z.xz)) \cup FV((\lambda x.(\lambda y.zxy))) \\ &= (FV(xz) \setminus \{z\}) \cup (FV(\lambda y.zxy) \setminus \{x\}) \\ &= (\{x, z\} \setminus \{z\}) \cup (FV(zxy) \setminus \{x, y\}) \\ &= \{x\} \cup (\{z, x, y\} \setminus \{x, y\}) \\ &= \{x, z\} \end{aligned}$$

As mentioned above, theories of the  $\lambda$ -calculus are deductive systems that describe the equations that hold between terms, the formulas of which are assertions  $M = N$ , for terms  $M$  and  $N$ . As one would expect, there are many different equivalence relations that may be defined on  $\Lambda$ . Hence there will be a number of possible theories, the strength and usefulness of which will depend on our choice of axioms governing the relation. We will be concerned here with a particularly strong theory, the  $\lambda\beta\eta$ -calculus, wherein all of the conventional axioms presented below are presumed to hold.

At the very least, all interesting lambda theories will satisfy the following minimal set of axiom schemes:

$$\begin{array}{c}
M = M \\
\\
\frac{M = N}{N = M} \\
\\
\frac{F = G \quad M = N}{FM = GN} \\
\\
\frac{M = N \quad N = P}{M = P} \\
\\
(\xi) \quad \frac{M = N}{\lambda x.M = \lambda x.N}
\end{array}$$

where  $M$ ,  $N$ ,  $F$ , and  $G$  are arbitrary terms in the language of the theory and  $x$  is any variable. Clearly, the first three axiom schemes merely require that “=” be an equivalence relation. The fourth specifies that equivalent terms are essentially indistinguishable with respect to their applicative behavior. The final axiom scheme, commonly referred to as  $(\xi)$ , states that if two terms  $M$  and  $N$  are equivalent, the terms corresponding to the assignments  $x \mapsto M$  and  $x \mapsto N$  are also equivalent. This property of the equivalence relation is often referred to as weak extensionality.

It is also customary to identify terms that are identical up to the renaming of bound variables. Since the distinction between such terms is a relatively vacuous one to begin with, this identification is often simply understood to hold. For the sake of clarity, however, we include it as an axiom, which we will call  $(\alpha)$ :

$$(\alpha) \quad \lambda x.M = \lambda y.M[y/x]$$

where  $y$  is a variable not appearing in  $M$ .

In addition, we include the axiom

$$(\beta) \quad (\lambda x.M)N = M[N/x] ,$$

with which we are already sufficiently acquainted.

Finally, we may wish to require that the equivalence relation have the property of extensionality, i.e. that terms which have the same effect on their arguments are equivalent. Formally, extensionality means that if a pair of terms  $M$  and  $N$  satisfy  $Mx = Nx$  for  $x \notin FV(MN)$ , then  $M = N$ . To ensure that the equivalence relation on terms is extensional, then, we might take this statement as an additional axiom, adjoining it to the set of axioms described above. As it turns out, though, we may obtain an equivalent theory by instead adjoining the simpler, more appealing axiom scheme  $(\eta)$ :

$$(\eta) \quad \lambda x.(Mx) = M,$$

for any variable  $x \notin FV(M)$ . The proof of the equivalence of the two theories, attributed to H. B. Curry, is included in [2].

As noted above, we are interested in the theory of the  $\lambda$ -calculus that has as its axioms all of the ones that have been described here, typically referred to as the  $\lambda\beta\eta$ -calculus. Although the introduction of types complicates matters, the theories of equations in the simply-typed  $\lambda$ -calculi that we will be studying are essentially the same as that of the  $\lambda\beta\eta$ -calculus.

## 1.2 The Simply-typed Calculus

At this point, it should be clear that although in many respects the terms of the type-free  $\lambda$ -calculus behave in a way that agrees with our intuitive ideas about functions, there are also respects in which it is impossible to draw a parallel between terms and functions. In particular, neither our common sense nor any conventional theory of functions would countenance the application of a function to itself, whereas this kind of self-applicative behavior is regarded as perfectly natural for type-free lambda terms. As suggested in the beginning of the chapter, the judicious assignment of types to terms, in conjunction with the imposition of certain natural restrictions on the operations, will prevent this from happening. In addition to the function types  $\sigma \rightarrow \tau$  described there, though, we will also include product types  $\sigma \times \tau$ . As we will see, the resulting typed  $\lambda$ -calculi will correspond to theories of functions that are considerably more comprehensible and more easily related to the others with which we are familiar.

It seems natural to begin with a description of the types involved. Intuitively, the types may be interpreted as sets, and statements of the form “ $M$  is a term of type  $\sigma$ ,” henceforth denoted  $M : \sigma$ , may be regarded as assertions of set membership. In this interpretation, the function type  $\sigma \rightarrow \tau$  represents the collection of functions from  $\sigma$  to  $\tau$ . It is clear, then, how the operation of application should be restricted so as to be compatible with the type structure: a term  $N : \sigma$  may serve as the argument of a term  $M$  if and only if  $M : \sigma \rightarrow \tau$  for some type  $\tau$ , in which case  $MN : \tau$ . We may also determine the way that the operation of abstraction behaves with respect to the type structure. Given a term  $M$  of type  $\tau$  (an element  $M$  of  $\tau$ ) and a variable  $x$  of type  $\sigma$  (a variable  $x$  that takes values in  $\sigma$ ), the expression  $\lambda x : \sigma.M$  describes a rule that assigns to each element of  $\sigma$  an element of  $\tau$ , meaning that  $(\lambda x : \sigma.M) : \sigma \rightarrow \tau$ . If we stopped at this point, we would be left with a minimal

typed theory, known as the pure typed  $\lambda$ -calculus, which incorporates only function types and the operations of application and abstraction. While theories of this kind are of considerable theoretical value, we will be interested in the significantly richer class of theories that incorporate product types  $\sigma \times \tau$ , as indicated above.

As suggested by the notation, the product types  $\sigma \times \tau$  are to be interpreted as the Cartesian products of the sets corresponding to  $\sigma$  and  $\tau$ . Although this is simple enough, the role that such types are to play in the theory may be somewhat mysterious. And well it should be: our discussion up to this point has not included any operations that could produce terms of type  $\sigma \times \tau$  or, indeed, that would allow us to distinguish the behavior of a term of type  $\sigma \times \tau$  from that of a term whose type is not a product. To make sense of the product types, we add three new operations on the terms. First, we introduce a pairing operation  $\langle -, - \rangle$ , which assigns to each pair  $M : \sigma$  and  $N : \tau$  a single term  $\langle M, N \rangle : \sigma \times \tau$ . We then add two associated operations,  $\pi_1(-)$  and  $\pi_2(-)$ , which correspond to the canonical projections in the first and second coordinates, respectively. In other words, if  $P$  is of type  $\sigma \times \tau$ , then  $\pi_1(P)$  is of type  $\sigma$  and  $\pi_2(P)$  is of type  $\tau$ .

Now that we have an understanding of the types and term forming operations, we may undertake a more formal description of the language of a typed theory  $T$  and of the process by which types are assigned to the terms. To that end, we make the following definition:

**Definition 1.2.1 (Types)** *The set of types of a theory  $T$  is freely generated over a countable family of basic types  $B_1, B_2, \dots$  according to the rule that if  $\sigma$  and  $\tau$  are types, then  $\sigma \times \tau$  and  $\sigma \rightarrow \tau$  are also types.*

The basic components from which we will build the language of  $T$ , denoted  $\mathcal{L}(T)$ , are as follows:

**Definition 1.2.2 (Alphabet)** *The alphabet of  $T$  consists of a countable family of variables  $X = \{x_1, x_2, \dots\}$ , a countable family of basic terms  $B = \{b_1, b_2, \dots\}$ , and the additional symbols  $\lambda$ ,  $(, )$ ,  $\langle$ , and  $\rangle$ .*

In any given theory, we will assume a fixed assignment of types to the basic terms, i.e.  $b_1 : \sigma_1, b_2 : \sigma_2, \dots$ , but will assume no such thing with respect to the variables. Since all of the terms of  $T$  are to be generated from the basic terms and variables, as we will soon see, the flexibility we have in assigning types to

variables will translate into a degree of variability in the types assigned to the more complicated terms. Specifically, the type that we assign to a term  $M$  will depend on the types assigned to its free variables. Hence our earlier informal assertions “ $M$  is of type  $\tau$ ” should be qualified and, instead, we should discuss type assignments only in relation to particular assignments of types to the relevant variables. This is accomplished through the introduction of the related notions of typing contexts, and of terms in context.

**Definition 1.2.3 (Context)** *A context  $\Gamma$  is a finite, possibly empty, set of type assignments to variables, i.e.  $\Gamma = \{x_1 : \sigma_1, x_2 : \sigma_2, \dots, x_n : \sigma_n\}$ , where the  $x_i$  are assumed to be distinct. We say that  $\Gamma$  is a context for a term  $M$  if every variable that occurs free in  $M$  appears in  $\Gamma$ . A term in context is an assertion of the form “ $M$  is of type  $\sigma$  with respect to  $\Gamma$ ,” denoted  $\Gamma \vdash M : \sigma$ .*

Using this new, more nuanced understanding, we may give a complete characterization of  $\mathcal{L}(T)$ , of the term operations, and of the rules by which types are to be assigned.

**Definition 1.2.4** *The language  $\mathcal{L}(T)$  is freely generated over the alphabet of  $T$  by the operations of application, abstraction, pairing, and projection, subject to the following rules:*

$$\begin{array}{c}
 \Gamma \vdash M : \tau \\
 \hline
 \Gamma, x : \sigma \vdash M : \tau \\
 \\
 \Gamma, x : \sigma \vdash M : \tau \qquad \Gamma \vdash M : \sigma \quad \Gamma \vdash F : \sigma \rightarrow \tau \\
 \hline
 \Gamma \vdash (\lambda x : \sigma. M) : \sigma \rightarrow \tau \qquad \Gamma \vdash F(M) : \tau \\
 \\
 \Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau \qquad \Gamma \vdash P : \sigma \times \tau \\
 \hline
 \Gamma \vdash \langle M, N \rangle : \sigma \times \tau \qquad \Gamma \vdash \pi_1(P) : \sigma \quad \Gamma \vdash \pi_2(P) : \tau
 \end{array}$$

With the exception of the “weakening” rule at the top right, which ensures that the type associated with a term in context is unchanged when additional typed variables are added to the context (provided that the new type assignments do not

contradict any preexisting ones), the rules outlined above are really just precise formulations of the intuitive ones that were laid out at the beginning of this section. For example, the rule

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x : \sigma. M) : \sigma \rightarrow \tau}$$

simply says that, with respect to a given context  $\Gamma$ , if  $M$  is of type  $\tau$  and  $x$  is a variable of type  $\sigma$ , then  $\lambda x : \sigma. M$  is of type  $\sigma \rightarrow \tau$ , just as we thought. At any rate, we now know exactly how to construct the terms of the language  $\mathcal{L}(T)$  of a typed  $\lambda$ -theory  $T$ , and we also know the rules according to which types are to be assigned to the terms.

Rather than redefine the notion of free variables that was introduced in section 1, we simply augment our earlier definition by a new clause stating that the set of free variables in a term of the form  $\langle M, N \rangle$  is given by  $FV(\langle M, N \rangle) = FV(M) \cup FV(N)$ . Notice that if a term  $M$  is closed, it will be assigned the same type  $\tau$  with respect to any context  $\Gamma$ , including the empty context. In this case, we write  $\vdash M : \sigma$ , and we may say unambiguously that  $M$  is of type  $\sigma$  in the theory. The only change in the process of substitution is that we now regard a term  $N$  as being substitutable for a free variable  $x$  in  $M$  only if  $N$  and  $x$  are of the same type, while retaining the condition that no free variables in  $N$  become bound when it is substituted for the free occurrences of  $x$  in  $M$ .

As in the case of the type-free theories examined in section 1, the theories of the typed  $\lambda$ -calculus will be deductive systems concerned with the equations that hold between terms. In typed calculi, however, the equations will be presumed to hold only between terms of the same type and, in particular, between terms in context. In other words, the assertions of the theory will be of the form  $\Gamma \vdash M = N$ , where  $\Gamma$  is a context for both  $M$  and  $N$ , with respect to which they are of the same type. To begin, every theory will include a (possibly empty) set of equations between closed terms  $M_1 = N_1, M_2 = N_2, \dots$ , which serves as a foundation for, and determines the nature of, the relation of syntactic equivalence. For each equation of this form, we write  $T \vdash M = N$ . As before, we also require a minimal set of axiom schemes governing the relation:

$$\begin{array}{c}
\Gamma \vdash M = M \\
\hline
\Gamma \vdash M = N \quad \Gamma \vdash N = P \\
\hline
\Gamma \vdash M = P \\
\hline
\Gamma \vdash F = G \quad \Gamma \vdash M = N \\
\hline
\Gamma \vdash F(M) = G(N)
\end{array}
\qquad
\begin{array}{c}
\Gamma \vdash M = N \\
\hline
\Gamma \vdash N = M \\
\hline
\Gamma \vdash M = N \\
\hline
\Gamma, x : \sigma \vdash M = N \\
\hline
\Gamma, x : \sigma \vdash M = N \\
\hline
(\xi) \quad \Gamma \vdash \lambda x : \sigma. M = \lambda x : \sigma. N
\end{array}$$

where the types of the terms  $F$ ,  $G$ ,  $M$ ,  $N$ , and  $P$  are such that the expressions above make sense. The attentive reader will find these axioms quite familiar since, with the exception of weakening rule represented by the fourth, they are merely the basic axioms of the type-free calculi translated into statements about terms in context. Once again, the first three axioms ensure that the relation “=” between terms in context is in fact an equivalence relation. The fourth says that we may add an extraneous typed variable to a context without affecting the equivalence of terms. The fifth ensures that if terms are equivalent in a given context, then their applicative behavior is the same. The sixth is the typed formulation of the principal of weak extensionality, which says that if  $M$  and  $N$  are equivalent terms of type  $\tau$  with respect to  $\Gamma, x : \sigma$ , then  $\lambda x : \sigma. M$  and  $\lambda x : \sigma. N$  are equivalent terms of type  $\sigma \rightarrow \tau$  with respect to  $\Gamma$ .

To ensure that the pairing and projection operations get along in the natural way, we add the following axiom schemes:

$$\begin{aligned}
\Gamma \vdash \pi_1(\langle M, N \rangle) &= M \\
\Gamma \vdash \pi_2(\langle M, N \rangle) &= N \\
\Gamma \vdash \langle \pi_1(P), \pi_2(P) \rangle &= P
\end{aligned}$$

for any terms  $M$ ,  $N$ , and  $P$ , where  $P$  is of some product type  $\sigma \times \tau$  with respect to  $\Gamma$ .

We also include the typed versions of the axioms  $(\alpha)$ ,  $(\beta)$ , and  $(\eta)$  that were introduced in Section 1. We begin with the typed analogue of  $(\alpha)$ , which equates terms that are identical up to the renaming of bound variables:

$$(\alpha) \quad \Gamma \vdash \lambda x : \sigma. M = \lambda y : \sigma. M[y/x]$$

where  $y$  is a variable not occurring in  $M$ . The new axiom

$$(\beta) \quad \Gamma \vdash (\lambda x : \sigma. M)N = M[N/x]$$

is essentially the same as the earlier one, although now the application  $(\lambda x : \sigma. M)N$  and the substitution of  $M$  for the free occurrences of  $x$  in  $N$  make sense only if  $N$  and  $x$  are both of type  $\sigma$ . Finally, we include

$$(\eta) \quad \Gamma \vdash \lambda x : \sigma. (Mx) = M$$

where  $\Gamma \vdash M : \sigma \rightarrow \tau$  for some  $\tau$  and  $x \notin FV(M)$ . As before,  $(\eta)$  guarantees that the relation of equivalence between terms in context is extensional, insofar as it will have the property that if  $M$  and  $N$  are terms of type  $\sigma \rightarrow \tau$  such that  $\Gamma \vdash Mx = Nx$  for all  $x$  of type  $\sigma$ , then  $\Gamma \vdash M = N$ .

In a given theory  $T$ , the basic equations and the rules described above generate an equivalence relation on the set of all closed terms of  $\mathcal{L}(T)$ , known as syntactic equivalence modulo  $T$ , which corresponds to provable equivalence within the theory. When closed terms  $M$  and  $N$  are equivalent in this sense, we write

$$T \vdash M = N$$

The goal of the present account is to explore the semantics of this logical calculus, as we will do momentarily.

### 1.3 Untyped as Typed

Before we proceed to other considerations, we would do well to consider a concrete example of a typed  $\lambda$ -theory, so as to be clear on the nature of the calculus. In particular, we will consider the theory of a reflexive domain which, in addition to providing a simple and comprehensible example, will also reveal the way in which an arbitrary type-free  $\lambda$ -theory may be regarded merely as a special part of a highly specialized kind of typed  $\lambda$ -theory. From this fact, of course, it will follow that the type-free calculus is a special case of the typed.

We begin by considering the theory of a reflexive domain, which we will denote by  $T$ . We define  $T$  to be a theory consisting of a single basic type  $U$ , a pair of

basic terms  $i : (U \rightarrow U) \rightarrow U$  and  $j : U \rightarrow (U \rightarrow U)$ , and a single basic equation relating them:  $\lambda x : (U \rightarrow U).ji(x) = \lambda x : (U \rightarrow U).x$ . In the categorical context, we would say that  $U$  retracts to the function object  $U \rightarrow U$ , insofar as the injection of  $U$  into the larger space  $U \rightarrow U$  is left inverse to the surjection of  $U \rightarrow U$  into  $U$ . Notice that we do not require that the equation  $\lambda x : U.ij(x) = \lambda x : U.x$  also hold (for the moment, at least). Hence  $U$  retracts, but is not isomorphic to,  $U \rightarrow U$ . We generate the types and terms of  $T$  from these basic types, terms, and a countable collection of variables  $X$  using the rules described above. Finally, we assume that the equivalence relation between terms in context is generated from the basic equation and the axioms laid out in Section 2. We may now prove the following:

**Proposition 1.3.1** *The theory  $T$  gives rise to a type-free  $\lambda\beta$ -calculus  $T'$ .*

**Proof:** First, construct the language  $\mathcal{L}$  of a type-free theory from the same countable collection of variables  $X$  used to generate  $\mathcal{L}(T)$ , but this time with no basic terms. We now define a translation  $-^\tau : \mathcal{L} \rightarrow \mathcal{L}(T)$ , which assigns to each term  $M$  of  $\mathcal{L}$  a term  $M^\tau$  of type  $U$  in  $\mathcal{L}(T)$ . Clearly, we will not be able to do this in the obvious way. If, for example, we defined the translation of an expression  $MN$  to be  $(MN)^\tau = M^\tau N^\tau$ , we would be left with an expression that consists of the application of a term of type  $U$  to another of type  $U$ , which cannot be a term of  $\mathcal{L}(T)$ . Similarly, if we defined  $(\lambda x.M)^\tau = \lambda x^\tau.M^\tau$ , the translation would be of type  $U \rightarrow U$  rather than  $U$ . To get around these problems, we make use of the basic terms  $i : (U \rightarrow U) \rightarrow U$  and  $j : U \rightarrow (U \rightarrow U)$ , defining the translation as follows:

1.  $x^\tau = x : U$
2.  $(MN)^\tau = j(M^\tau)N^\tau$
3.  $(\lambda x.M)^\tau = i(\lambda x^\tau.M^\tau)$

Define an equivalence relation on the terms of  $\mathcal{L}$  by the condition that  $M = N$  if and only if  $M^\tau = N^\tau$  in the typed theory. I claim that the relation thus defined satisfies almost all of the axioms described in Section 1, with the extensionality property codified by  $(\eta)$  as the only possible extension. Hence the resulting theory,  $T'$ , will be a  $\lambda\beta$ -calculus. The verification of the basic axioms is a trifling business, so we need only consider  $(\alpha)$  and  $(\beta)$ . Notice that, since we assign type  $U$  to every variable, the contexts in the following computations are essentially trivial, and they may therefore be omitted.

To prove that  $(\alpha)$  holds, i.e. that  $\lambda x.M = \lambda y.M[y/x]$  for all  $y$  not appearing in  $M$ , we must show that  $(\lambda x.M)^\tau = \lambda y.M[y/x]$ . But this is a simple matter:

$$\begin{aligned}
(\lambda x.M)^\tau &= i(\lambda x^\tau.M^\tau) \\
&\stackrel{\alpha}{=} i(\lambda y^\tau.M^\tau[y^\tau/x^\tau]) \quad (\text{for } y^\tau \text{ not in } M^\tau) \\
&= i(\lambda y^\tau.(M[y/x])^\tau) \\
&= (\lambda y.M[y/x])^\tau
\end{aligned} \tag{1.1}$$

where (1.1) follows from the fact that the typed calculus  $T$  satisfies the typed analogue of  $(\alpha)$ , and since  $y^\tau$  appears in  $M^\tau$  if and only if  $y$  appears in  $M$ ,  $T'$  satisfies  $(\alpha)$ .

The verification that  $T'$  obeys  $(\beta)$  is similar, although in this case we use the basic equation relating the terms  $i$  and  $j$ :

$$\begin{aligned}
((\lambda x.M)N)^\tau &= j((\lambda x.M)^\tau)N^\tau \\
&= ji(\lambda x^\tau.M^\tau)N^\tau \\
&= ((\lambda x^\tau.ji(x^\tau))(\lambda x^\tau.M^\tau))N^\tau \\
&= ((\lambda x^\tau.x^\tau)(\lambda x^\tau.M^\tau))N^\tau \\
&= (\lambda x^\tau.M^\tau)N^\tau \\
&= M^\tau[N^\tau/x^\tau] \\
&= (M[N/x])^\tau
\end{aligned} \tag{1.2}$$

$$\tag{1.3}$$

where equations (1.2) and (1.3) hold by virtue of the fact that  $T$  satisfies  $(\beta)$ . Hence  $T'$  is a  $\lambda\beta$ -calculus, as claimed.  $\square$

The theory of a reflexive domain is of considerable interest in itself, and will also have significant ramifications in our search for semantics. Still, one can see that the type-free theories  $T'$  that are produced in the above manner need not be extensional, even if the underlying typed theory  $T$  is. It seems natural to inquire, then, what further restrictions we may place on  $T$  in order to ensure that  $T'$  is not merely a  $\lambda\beta$ -calculus, but rather a  $\lambda\beta\eta$ -calculus. As it turns out, it suffices to include  $\lambda x : U.ij(x) = x$  as an additional basic equation of  $T$ . Given that  $T$  has this

as a basic equation and also satisfies  $(\eta)$ , we may calculate that

$$\begin{aligned}
(\lambda x.(Mx))^\tau &= i(\lambda x^\tau.(Mx)^\tau) \\
&= i(\lambda x^\tau.(j(M^\tau)x^\tau)) \\
&\stackrel{\eta}{=} i(j(M^\tau)) \\
&\stackrel{\beta}{=} (\lambda x^\tau.ij(x^\tau))M^\tau \\
&= (\lambda x^\tau.x^\tau)M^\tau \\
&= M^\tau
\end{aligned}$$

Of course, this means that  $\lambda x.(Mx) = M$  in  $T'$ , i.e. that  $T'$  is extensional.

We have seen, then, that any theory of a reflexive domain gives rise to a type-free  $\lambda\beta$ - or  $\lambda\beta\eta$ -calculus, depending on the equations that hold between the basic terms  $i$  and  $j$ . Naturally, the interpretation above may be extended to theories of a reflexive domain that include additional basic terms of type  $U$  in the obvious way, thereby yielding type-free theories with the corresponding basic terms. Considerations of this kind may lead one to suspect the punchline of this section: there is a kind of converse to the construction described above. In particular, given any type-free theory, there is a typed theory (of a reflexive domain) such that the interpretation above yields precisely the same type-free theory. The construction by which we obtain the relevant typed theory, due to Dana Scott [15], is too involved to be presented here but it should seem plausible, at the very least, that type-free  $\lambda$ -calculi correspond to theories of reflexive domains. In this sense, the typed calculus may be regarded as the more general theory, and the type-free calculus as a special case.

## Chapter 2

# Elementary Categorical Semantics

Now that we possess an adequate understanding of the syntax of the typed  $\lambda$ -calculus, we may turn our attention to its semantics, the heart of the present inquiry. The ultimate aim, of course, is to establish that every typed  $\lambda$ -theory  $T$  has a model in a particular subcategory of a slice category of the form  $\mathbf{Pos}/\mathcal{O}(X_T)$ , where  $\mathbf{Pos}$  denotes the category of partially ordered sets (posets) and  $\mathcal{O}(X_T)$  is the poset of open subsets of some topological space  $X_T$ . Moreover, we will show that the model of  $T$  in this subcategory (which consists of the fibrations over  $\mathcal{O}(X_T)$ ) is a representation of  $T$ , in the sense that the types and terms of  $T$  are interpreted, respectively, as objects and arrows in the subcategory, every arrow in the subcategory is the interpretation of a term of  $T$ , and terms of  $T$  have the same interpretation if and only if they are syntactically equivalent. The goal of this chapter is more modest, insofar as we aim only to set out a few of the basic results that will serve as a foundation for the arguments in the ensuing chapters. In particular, it will be shown in Section 1 that given any  $\lambda$ -theory  $T$ , we may construct its free syntactic category  $\mathbf{C}_T$ , a cartesian closed category in which the objects are the types of  $T$  and arrows between objects  $\sigma$  and  $\tau$  correspond to equivalence classes of closed terms of type  $\sigma \rightarrow \tau$  in  $T$ . As we will see in Section 2, the freeness of this construction will imply, among other things, that cartesian closed categories provide complete and strongly complete semantics and, moreover, that every lambda theory has a representation in this system of semantics. The syntactic category construction will also provide us with a way of extending the completeness results of Section 1 to arbitrary categories by considering the functorial images of the representation. This result is applied immediately in Section 3, as we use the Yoneda embedding of a category into its presheaves (the contravariant set-valued functors defined on the category) to prove

that every lambda theory has a presheaf representation. Before we become immersed in the details, though, it seems appropriate to introduce the basic terms used in connection with the semantics of the typed  $\lambda$ -calculus, and to briefly consider the present state of our knowledge in this field.

As in [1], we begin by defining the basic semantic notions. First, given a  $\lambda$ -theory  $T$  and an arbitrary category  $\mathbf{C}$ , we define an interpretation of  $T$  in  $\mathbf{C}$  to be a map  $\llbracket - \rrbracket : T \rightarrow \mathbf{C}$  that assigns to each type  $\sigma$  an object  $\llbracket \sigma \rrbracket$  of  $\mathbf{C}$ , and assigns to each term in context  $\Gamma \vdash M : \sigma$  an arrow  $\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket \sigma \rrbracket$  in  $\mathbf{C}$  (we will make this more precise in a moment). As one would expect, a model of  $T$  in  $\mathbf{C}$  is a special kind of interpretation. Specifically,

**Definition 2.0.1** *A model  $\mathcal{M}$  of  $T$  in  $\mathbf{C}$ , represented schematically as  $\mathcal{M} : T \rightarrow \mathbf{C}$ , is an interpretation  $\llbracket - \rrbracket_{\mathcal{M}}$  of  $T$  in  $\mathbf{C}$  such that for any terms  $M$  and  $N$  in  $\mathcal{L}(T)$ ,  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$  whenever  $T \vdash M = N$ . We say that a model  $\mathcal{M} : T \rightarrow \mathbf{C}$  is standard if  $\mathbf{C}$  is a cartesian closed category and the function types  $\sigma \rightarrow \tau$  are interpreted as exponential objects in the category, i.e.  $\llbracket \sigma \rightarrow \tau \rrbracket_{\mathcal{M}} = \llbracket \tau \rrbracket_{\mathcal{M}}^{\llbracket \sigma \rrbracket_{\mathcal{M}}}$ . Otherwise, we say that  $\mathcal{M}$  is nonstandard.*

Of course, we are not merely interested in semantics for particular  $\lambda$ -theories, but rather in semantics for typed  $\lambda$ -theories in general. Hence we introduce the following concept:

**Definition 2.0.2** *A system of semantics for the typed  $\lambda$ -calculus is a collection  $S$  of models  $\mathcal{M} : T \rightarrow \mathbf{C}$  for each  $\lambda$ -theory  $T$ , where the category  $\mathbf{C}$  is not necessarily the same for all  $\mathcal{M}$  in  $S$ . The collection of models in a particular category  $\mathbf{C}$  is referred to as a  $\mathbf{C}$ -valued semantics. When all of the models in  $S$  are standard, we say that  $S$  is a standard system of semantics.*

We may now give a formal characterization of the notions of completeness that will be used in this account:

**Definition 2.0.3** *We say that a system of semantics  $S$  is complete if for every  $\lambda$ -theory  $T$  and terms  $M$  and  $N$  in  $\mathcal{L}(T)$ ,*

$$T \vdash M = N \quad \text{if and only if} \quad \llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}} \text{ for all } \mathcal{M} \in S.$$

*In other words,  $S$  is complete if the calculus of syntactic equivalence described in Section 1.2 is sound and complete with respect to  $S$ . We say that  $S$  is strongly complete if for every  $\lambda$ -theory  $T$  there exists a model  $\mathcal{M} \in S$  such that*

$$T \vdash M = N \quad \text{if and only if} \quad \llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}},$$

in which case we say that  $\mathcal{M}$  is a complete model for  $T$ .

Finally,

**Definition 2.0.4** *Given a standard model  $\mathcal{M} : T \rightarrow \mathbf{C}$ , we say that  $\mathcal{M}$  is functionally complete if every  $\mathbf{C}$ -arrow of the form  $m : 1 \rightarrow \llbracket \sigma \rrbracket_{\mathcal{M}}$  is the interpretation of a closed term  $M : \sigma$  in  $\mathcal{L}(T)$ , i.e.  $m = \llbracket M \rrbracket_{\mathcal{M}}$ . As we will soon see, this condition will also imply that every  $\mathbf{C}$ -arrow  $f : \llbracket \sigma \rrbracket_{\mathcal{M}} \rightarrow \llbracket \tau \rrbracket_{\mathcal{M}}$  is the interpretation of a term in context of the form  $x : \sigma \vdash F[x] : \tau$ , where  $x$  is the only free variable in  $F[x]$ .*

Using this terminology, we may restate the definition of a representation of a  $\lambda$ -theory  $T$ , which was hinted at in the introduction to this chapter.

**Definition 2.0.5** *We say that a model  $\mathcal{M} : T \rightarrow \mathbf{C}$  is a representation if it is standard, complete, and functionally complete.*

In other words,  $\mathcal{M}$  is a representation if the function types are interpreted as exponentials, every arrow  $f : 1 \rightarrow \llbracket \sigma \rrbracket_{\mathcal{M}}$  in  $\mathbf{C}$  is the interpretation of a closed term  $M : \sigma$  and, by completeness, this term is unique up to provable equivalence in  $T$ , i.e. up to syntactic equivalence.

In the last half century, the possible systems of semantics for the typed  $\lambda$ -calculus, and the completeness thereof, have been the subject of a considerable amount of research, beginning with Henkin's proof of the completeness of a non-standard set-valued semantics ([6]). In essence, the models that Henkin used were based on ordered pairs of the form  $(\{A^\sigma\}, I)$ , where  $\{A^\sigma\}$  is a family of sets indexed by the types of the theory (with  $A^\sigma$  being thought of as the set of terms of type  $\sigma$ ), and  $I$  is an interpretation of the basic terms  $b : \sigma$  in the appropriate  $A^\sigma$ . The important thing to note is that in such models one requires only that  $A^{\sigma \rightarrow \tau}$  be a subset of the set  $(A^\tau)^{A^\sigma}$  of mappings from  $A^\sigma$  to  $A^\tau$ , and not that  $A^{\sigma \rightarrow \tau} = (A^\tau)^{A^\sigma}$  as we would expect in a standard model. Clearly, the class of Henkin models is larger than the class of standard models in the category of sets and, as a result, the completeness of Henkin's system of semantics cannot be translated into a similar result for standard set-valued semantics.

Indeed, while much of the recent work in the field has been concerned with establishing that standard set-valued semantics is complete for certain limited classes

of  $\lambda$ -theories (theories with a single basic type and no equations or constants, theories with constants but no equations, and so on), one can readily see that standard semantics in the category of sets cannot be complete for  $\lambda$ -theories in general. Consider the theory of a reflexive domain considered in Section 1.3, for example, which has a single basic type  $U$ , basic terms  $i : (U \rightarrow U) \rightarrow U$  and  $j : U \rightarrow (U \rightarrow U)$ , and the basic equation  $\vdash \lambda x : (U \rightarrow U).ji(x) = \lambda x : (U \rightarrow U).x$ . If it has a model in the category of sets, there must be a set  $U$  and set mappings  $i : U^U \rightarrow U$  and  $j : U \rightarrow U^U$  such that for all  $y \in U^U$ ,  $ji(y) = y$ . But this implies that for every  $y \in U^U$  there is an element  $i(y) \in U$  such that  $y = j(i(y))$ . Hence the cardinality of  $U$  must be greater than or equal to that of  $U^U$ ,  $|U| \geq |U^U|$ . Since the cardinality of the set  $U^U$  of maps from  $U$  to itself must be at least as large as that of  $U$ , it follows that  $|U| = |U^U|$ . Of course, this can happen only when  $|U| = |U^U| = 1$ , in which case  $U$  is a singleton. But if  $U$  is a singleton in every model of the theory, then the equation  $ij(x) = x$  holds for all  $x \in U$  in every model, whereas  $\lambda x : U.ij(x) = \lambda x : U.x$  is not derivable in the theory. Naturally, this means that standard semantics in **Set**, the category of sets, is not complete by the definition above.

As suggested in the introduction to this section, though, we will see momentarily that the class of standard models in cartesian closed categories constitutes a complete and strongly complete system of semantics for the typed  $\lambda$ -calculus and, furthermore, that it contains a representation of every  $\lambda$ -theory. This result is readily extended to standard semantics in presheaf categories of the form **Set**<sup>**C**op</sup>, where **C** is cartesian closed. Using the completeness of presheaf semantics, in conjunction with a recent result from topos theory, Steven Awodey ([1]) has proved that every  $\lambda$ -theory  $T$  has a representation in the category of sheaves on a topological space  $X_T$ , as we will see in Chapter 3. As a trivial corollary to this theorem, we show that  $T$  also has a representation in the category of presheaves on the poset of open subsets of  $X_T$ , denoted  $\mathcal{O}(X_T)$ . While this result is, for our purposes, merely an intermediate step in the proof of the fibration representation theorem, it is of some interest in itself. Whereas Mitchell and Moggi ([14]) have established the strong completeness of a system of semantics consisting of nonstandard “Kripke-style” models in categories of the form **Set**<sup>**P**</sup>, where **P** is a poset, we will prove an analogous result for standard models in the smaller class of categories where **P** is the poset of open subsets of a topological space. As we will see in Chapter 4, the fibration representation theorem is a consequence of this improved presheaf completeness result, and has the virtue of replacing the complicated functor categories mentioned above with

relatively straightforward categories of posets and monotone functions.

## 2.1 Lambda Theories, Cartesian Closed Categories

We now turn our attention to the relationship between typed  $\lambda$ -theories and cartesian closed categories. On the basis of the discussion in the introduction and the first chapter, it would seem that we have some reason to believe that the correspondence between the two is very close indeed. As we have seen, a typed  $\lambda$ -theory may be interpreted as a theory of functions, consisting of a collection of domains (the types) and functions (the terms), subject to operations of application, pairing, and abstraction (or currying), each of which is determined by the corresponding operation in the  $\lambda$ -calculus. A cartesian closed category may be interpreted as theory of functions subject to precisely the same operations, where the domains are now the objects of the category, the functions are the arrows, and the operations in the theory are determined by the categorical operations of composition, pairing, and transposition. In some sense, then, we might associate a given  $\lambda$ -theory  $T$  with a cartesian closed category  $\mathbf{C}$  if  $T$  and  $\mathbf{C}$  give rise to the same theory of functions.

There is a more direct and formal correspondence, however, a categorical equivalence which is by now well known among category theorists. To see how this works, we form the category of  $\lambda$ -calculi, denoted  $\lambda\text{-}\mathbf{Calc}$ , whose objects are  $\lambda$ -calculi and whose arrows are “transformations” between  $\lambda$ -calculi that preserve the type structure, operations, and assignments in the natural way. Similarly, we may form  $\mathbf{Cart}$ , the category of all cartesian closed categories, the morphisms of which are functors that preserve the cartesian closed structure. The categories  $\lambda\text{-}\mathbf{Calc}$  and  $\mathbf{Cart}$  are equivalent in the sense that there are functors  $C : \lambda\text{-}\mathbf{Calc} \rightarrow \mathbf{Cart}$  and  $L : \mathbf{Cart} \rightarrow \lambda\text{-}\mathbf{Calc}$  such that for any  $\lambda$ -theory  $T$  and cartesian closed category  $\mathbf{C}$ ,

$$L(C(T)) \cong T \quad \text{and} \quad C(L(\mathbf{C})) \cong \mathbf{C}$$

The equivalence itself, as well as the functor  $L : \mathbf{Cart} \rightarrow \lambda\text{-}\mathbf{Calc}$  (which assigns to each category the typed  $\lambda$ -theory known as its internal language), are of fundamental importance in the field of categorical logic. As a result, they have rightly been given a great deal of attention, particularly in [9]. Still, while it is good to keep this general framework in mind, these topics fall outside the scope of the present inquiry. For our purposes, it suffices to consider the functor  $C : \lambda\text{-}\mathbf{Calc} \rightarrow \mathbf{Cart}$  which, as suggested above, assigns to each  $\lambda$ -theory  $T$  its free syntactic category.

Before we proceed, though, recall that a category  $\mathbf{C}$  is said to be cartesian closed if it satisfies the following conditions:

1.  $\mathbf{C}$  has a terminal object  $1$ , which has the property that for every object  $A$  there is a unique arrow  $!_A : A \rightarrow 1$ .
2. For any objects  $A$  and  $B$ , there is an object  $A \times B$  and a pair of projections  $p_1 : A \times B \rightarrow A$  and  $p_2 : A \times B \rightarrow B$  with the property that for any  $f : X \rightarrow A$  and  $g : X \rightarrow B$  there is a unique  $(f, g) : X \rightarrow A \times B$  such that the following diagram commutes:

$$\begin{array}{ccccc}
 & & X & & \\
 & \swarrow & \downarrow & \searrow & \\
 & f & (f, g) & g & \\
 A & \xleftarrow{p_1} & A \times B & \xrightarrow{p_2} & B
 \end{array}$$

Or, written equationally,

$$\begin{aligned}
 p_1(f, g) &= f \\
 p_2(f, g) &= g
 \end{aligned}$$

It follows easily from the definition that for any  $h : X \rightarrow A \times B$ ,

$$(p_1 \circ h, p_2 \circ h) = h$$

One can see, then, that pairing induces a bijection between Hom-sets:

$$\text{Hom}_{\mathbf{C}}(X, A) \times \text{Hom}_{\mathbf{C}}(X, B) \cong \text{Hom}_{\mathbf{C}}(X, A \times B)$$

3. For any objects  $A$  and  $B$ , there is an object  $B^A$  (regarded informally as the collection of morphisms from  $A$  to  $B$ ) and an evaluation arrow  $\epsilon : B^A \times A \rightarrow B$  with the property that for any arrow  $f : C \times A \rightarrow B$  there is a unique transpose  $\tilde{f} : C \rightarrow B^A$  such that the following diagram commutes:

$$\begin{array}{ccccc}
 & & B^A & & \\
 & & \vdots & & \\
 & & \tilde{f} & & \\
 & & \vdots & & \\
 & & C & & \\
 & & \uparrow & & \\
 & & C \times A & & \\
 & \swarrow & \uparrow & \searrow & \\
 & & B^A \times A & \xrightarrow{\epsilon} & B \\
 & & (\tilde{f} \circ p_1, p_2) & & \\
 & & \uparrow & & \\
 & & C \times A & \xrightarrow{f} & B
 \end{array}$$

In other words,

$$f = \epsilon(\tilde{f} \circ p_1, p_2).$$

Trivially, each  $g : C \rightarrow B^A$  determines a unique arrow  $\tilde{g} = \epsilon(g \circ p_1, p_2) : C \times A \rightarrow B$ , called the transpose of  $g$ , such that the equation above is satisfied when  $\tilde{f}$  is replaced by  $g$  and  $f$  is replaced by  $\tilde{g}$ . Thus we have

$$g = \epsilon(g \circ p_1, p_2) = \tilde{g},$$

whence it follows that for all  $f : C \times A \rightarrow B$ ,  $f = \tilde{f}$ . Hence transposition induces a bijection between Hom-sets:

$$\text{Hom}_{\mathbf{C}}(C \times A, B) \cong \text{Hom}_{\mathbf{C}}(C, B^A).$$

We now form the category **Cart**, which has as its objects all cartesian closed categories. The morphisms in **Cart** are functors that preserve the cartesian closed structure “on the nose.” In other words,  $F : \mathbf{C} \rightarrow \mathbf{D}$  is an arrow in **Cart** if for any objects  $A$  and  $B$  and arrows  $f$  and  $g$  in  $\mathbf{C}$ ,

$$\begin{aligned} F(1) &= 1, & F(A \times B) &= F(A) \times F(B), & F(B^A) &= F(B)^{F(A)}, \\ F(!_A) &= !_A, & F((f, g)) &= (F(f), F(g)), & & \text{etc.} \end{aligned}$$

Functors of this kind are said to be cartesian closed. As suggested above, **Cart** is to be the codomain of the functor  $C$ , which we now consider. In particular, we examine its object function  $C : T \mapsto \mathbf{C}_T$ .

It is to be hoped that the brief sketch of the properties of cartesian closed categories presented above has made clear the nature of the analogy between the types, terms and operations in a  $\lambda$ -theory and the objects, arrows, and operations in a cartesian closed category, as it is this analogy that will guide us in the construction of  $\mathbf{C}_T$ . We ought to be careful, though, since, on the basis of our understanding of the  $\lambda$ -calculus, we have no reason to suspect that a  $\lambda$ -theory  $T$  has any structure resembling a terminal object. To complete the correspondence, we add a new basic type  $1$ , a new basic term  $* : 1$ , and a basic equation  $x : 1 \vdash x = *$ . It is easy to see that these additions will have no effect on the relation of syntactic equivalence between the preexisting terms, i.e. that for any terms  $M$  and  $N$  in  $\mathcal{L}(T)$  generated from the original families of basic types and terms, it will be the case that  $T \vdash M = N$  after the additions if and only if  $T \vdash M = N$  held beforehand. Hence

we may safely make the suggested changes. Now, the category  $\mathbf{C}_T$  is constructed as follows:

The objects of  $\mathbf{C}_T$  are the types of  $T$ .

The arrows between any objects  $\sigma$  and  $\tau$  are closed terms  $M : \sigma \rightarrow \tau$ , identified up to syntactic equivalence. In other words, if  $T \vdash M = N$ , then  $M$  and  $N$  define the same arrow from  $\sigma$  to  $\tau$ . Equality of arrows in the category is defined by the condition that the arrows corresponding to closed terms  $M$  and  $N$  are equal if and only if  $T \vdash M = N$ .

To ensure that  $\mathbf{C}_T$  is a well-defined category and, in addition, that it is cartesian closed, we must also identify the terms of  $\mathcal{L}(T)$  that correspond to the distinguished arrows and operations in a cartesian closed category. In particular, we must specify the form of the identity, projection, and evaluation arrows, as well as the operations of composition, pairing, and transposition. We do this as follows:

$$\begin{aligned}
1_\sigma &= \lambda x.x && (x : \sigma) \\
f \circ g &= \lambda x.f(g(x)) \\
p_1 &= \lambda z.\pi_1(z) && (z : \sigma \times \tau) \\
p_2 &= \lambda z.\pi_2(z) && (z : \sigma \times \tau) \\
(f, g) &= \lambda x.\langle fx, gx \rangle \\
\epsilon &= \lambda z.(\pi_1(z)\pi_2(z)) && (z : ((\sigma \rightarrow \tau) \times \sigma) \rightarrow \tau) \\
h^* &= \lambda xy.h\langle x, y \rangle && (x : \sigma_1, y : \sigma_2)
\end{aligned}$$

where  $f$ ,  $g$ , and  $h$  are arrows in  $\mathbf{C}_T$  such that these expressions make sense.

Although the verification that  $\mathbf{C}_T$  has all of the properties required of a cartesian closed category is relatively simple, it cannot hurt to check that  $\mathbf{C}_T$  satisfies a few of the necessary equations. For example, we may show that the identity arrows behave in the correct way: if  $f : \sigma \rightarrow \tau$ ,

$$1_\tau \circ f = \lambda x.(1_\tau(fx)) = \lambda x.((\lambda y.y)(fx)) = \lambda x.(fx) = f$$

$$f \circ 1_\sigma = \lambda x.(f(1_\sigma x)) = \lambda x.(f((\lambda y.y)x)) = \lambda x.(fx) = f$$

To check that  $\mathbf{C}_T$  is a category (at the very least), it now suffices to show that the

operation of composition is associative. But clearly, if  $f$ ,  $g$ , and  $h$  are composable,

$$\begin{aligned}
f \circ (g \circ h) &= \lambda x.(f((g \circ h)x)) \\
&= \lambda x.(f((\lambda y.g(hy))x)) \\
&= \lambda x.(f(g(hx))) \\
&= \lambda x.((\lambda y.f(gy))(hx)) \\
&= \lambda x.((f \circ g)(hx)) \\
&= (f \circ g) \circ h
\end{aligned}$$

Of course, one must now verify that  $\mathbf{C}_T$  satisfies the equations involving pairing, projection, transposition and evaluation. Instead of going through the arguments here, however, we check that  $p_1(f, g) = f$  and assume the rest, leaving the proofs as exercises for the skeptical reader. If  $f$  and  $g$  are arrows such that the expression  $(f, g)$  makes sense,

$$\begin{aligned}
p_1(f, g) &= \lambda x.(p_1((f, g)x)) \\
&= \lambda x.((\lambda y.\pi_1(y))((f, g)x)) \\
&= \lambda x.(\pi_1((f, g)x)) \\
&= \lambda x.(\pi_1((\lambda y.\langle fy, gy \rangle)x)) \\
&= \lambda x.(\pi_1(\langle fx, gx \rangle)) \\
&= \lambda x.(fx) \\
&= f
\end{aligned}$$

The syntactic category thus constructed has another description, though, which is more useful in the present context. First, recall that the language  $\mathcal{L}(T)$  of a typed  $\lambda$ -theory  $T$  is freely generated over a family of basic types and terms using the by now familiar  $\lambda$ -calculus operations, modulo the equivalence relation generated by the axioms and basic equations. Given the close relationship between the types and terms of  $T$  and the objects and arrows of  $\mathbf{C}_T$ , and between the operations in  $T$  and  $\mathbf{C}_T$ , it is clear that we may view the syntactic category as being the cartesian closed category freely generated over a family of basic objects and arrows. Specifically, if  $B_1, B_2, \dots$  and  $b_1 : \sigma_1, b_2 : \sigma_2, \dots$  are the basic types and terms of  $T$ ,  $\mathbf{C}_T$  is the cartesian closed category freely generated over the same collections, where  $B_1, B_2, \dots$  are interpreted as objects, basic terms  $b : \sigma \rightarrow \tau$  are interpreted as arrows between objects  $\sigma$  and  $\tau$ , and basic terms  $b : \sigma$  (where  $\sigma$  is not a function type) are interpreted as arrows between 1 and  $\sigma$ .

As we will soon see, the freeness of this construction yields important results. In particular, every  $\lambda$ -theory  $T$  has a representation  $\mathcal{U}$  in its syntactic category which has the property that for any cartesian closed category  $\mathbf{C}$ , models  $\mathcal{M} : T \rightarrow \mathbf{C}$  correspond uniquely to cartesian closed functors  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$  such that  $\mathcal{M}^*(\mathcal{U}) = \mathcal{M}$ . This fundamental fact makes it possible to analyze the completeness of models in arbitrary cartesian closed categories in terms of the properties of the corresponding functors from the syntactic category, a method by which we will quickly obtain important results.

## 2.2 Models In Arbitrary CCCs

As the reader will no doubt recall, our earlier presentation of the notion of an interpretation of a  $\lambda$ -theory  $T$  in an arbitrary category  $\mathbf{C}$  was rather vague. Before we proceed with our study of the semantics of the  $\lambda$ -calculus, then, we require a more precise definition. When  $\mathbf{C}$  is cartesian closed, this may be accomplished in a particularly nice way.

An interpretation of a typed  $\lambda$ -theory  $T$  in  $\mathbf{C}$  is a map  $\llbracket - \rrbracket : T \rightarrow \mathbf{C}$ , which maps each basic type  $B$  of  $T$  to an object  $\llbracket B \rrbracket$  of  $\mathbf{C}$  and each basic term  $b : \sigma$  to an arrow  $\llbracket b \rrbracket : 1 \rightarrow \llbracket \sigma \rrbracket$ . The map  $\llbracket - \rrbracket$  is extended inductively to all types according to the rules  $\llbracket \sigma \times \tau \rrbracket = \llbracket \sigma \rrbracket \times \llbracket \tau \rrbracket$  and  $\llbracket \tau^\sigma \rrbracket = \llbracket \tau \rrbracket^{\llbracket \sigma \rrbracket}$ . In addition,  $\llbracket - \rrbracket$  is extended to arbitrary terms in context according to the following rules:

1.  $\llbracket x : \sigma \vdash x : \sigma \rrbracket = 1_{\llbracket \sigma \rrbracket}$
2. If  $x : \sigma$  is not free in  $M$ ,

$$\llbracket \Gamma, x : \sigma \vdash M : \tau \rrbracket = \llbracket \Gamma \vdash M : \tau \rrbracket \circ p$$

where  $p$  is the projection  $\llbracket \Gamma \rrbracket \times \llbracket \sigma \rrbracket \rightarrow \llbracket \Gamma \rrbracket$ .

3.  $\llbracket \Gamma \vdash (\lambda x : \sigma. M) : (\sigma \rightarrow \tau) \rrbracket = \llbracket \Gamma, x : \sigma \vdash M : \tau \rrbracket^\sim$ , the transpose of  $\llbracket \Gamma, x : \sigma \vdash M : \tau \rrbracket$ .
4.  $\llbracket \Gamma \vdash \langle M, N \rangle : (\sigma \times \tau) \rrbracket = (\llbracket \Gamma \vdash M : \sigma \rrbracket, \llbracket \Gamma \vdash N : \tau \rrbracket)$
5.  $\llbracket \Gamma \vdash \pi_1(P) : \sigma \rrbracket = p_1 \circ \llbracket \Gamma \vdash P : (\sigma \times \tau) \rrbracket$  and  $\llbracket \Gamma \vdash \pi_2(P) : \sigma \rrbracket = p_2 \circ \llbracket \Gamma \vdash P : (\sigma \times \tau) \rrbracket$ , where  $p_1 : \llbracket \sigma \rrbracket \times \llbracket \tau \rrbracket \rightarrow \llbracket \sigma \rrbracket$  and  $p_2 : \llbracket \sigma \rrbracket \times \llbracket \tau \rrbracket \rightarrow \llbracket \tau \rrbracket$  are the canonical projections.

6.  $\llbracket \Gamma \vdash MN : \tau \rrbracket = \epsilon \circ \llbracket \Gamma \vdash \langle M, N \rangle : ((\sigma \rightarrow \tau) \times \sigma) \rrbracket$ , where  $\epsilon : \llbracket \tau \rrbracket^{\llbracket \sigma \rrbracket} \times \llbracket \sigma \rrbracket \rightarrow \llbracket \tau \rrbracket$  is the evaluation arrow.

Every context  $\Gamma = \{x_1 : \sigma_1, x_2 : \sigma_2, \dots, x_n : \sigma_n\}$  is interpreted as  $\llbracket \Gamma \rrbracket = \llbracket \sigma_1 \rrbracket \times \llbracket \sigma_2 \rrbracket \times \dots \times \llbracket \sigma_n \rrbracket$ . The empty context is interpreted as the empty product, i.e. the terminal object 1. Hence every closed term  $\vdash M : \sigma$  is interpreted as an arrow  $\llbracket M \rrbracket : 1 \rightarrow \llbracket \sigma \rrbracket$ . Naturally, a model  $\mathcal{M} : T \rightarrow \mathbf{C}$  is an interpretation of  $T$  in  $\mathbf{C}$  such that for any terms  $M$  and  $N$ ,  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$  whenever  $T \vdash M = N$ .

It should be clear from the construction of the syntactic category that there will be at least one model of  $T$  in  $\mathbf{C}_T$ . In particular, there is a model  $\mathcal{U} : T \rightarrow \mathbf{C}_T$  such that for any basic type  $B$ ,  $\llbracket B \rrbracket_{\mathcal{U}} = B$  (regarded as an object of  $\mathbf{C}_T$ ), and for any basic term  $b : \sigma \rightarrow \tau$ ,  $\llbracket b \rrbracket_{\mathcal{U}} = \tilde{b} : 1 \rightarrow \llbracket \tau \rrbracket_{\mathcal{U}}^{\llbracket \sigma \rrbracket_{\mathcal{U}}}$  (where  $b$  is regarded as an object of  $\mathbf{C}_T$ ). Of course if  $b : \sigma$ , with  $\sigma$  not a function type, then  $\llbracket b \rrbracket_{\mathcal{U}} = b : 1 \rightarrow \sigma$ . When we extend the interpretation according to the rules above, we see that closed terms  $M : \sigma \rightarrow \tau$  are interpreted as  $\llbracket M \rrbracket_{\mathcal{U}} = \tilde{M} : 1 \rightarrow \tau^{\sigma}$ , terms in context  $x : \sigma \vdash N : \tau$  are interpreted as arrows  $\llbracket N \rrbracket_{\mathcal{U}} : \llbracket \sigma \rrbracket_{\mathcal{U}} \rightarrow \llbracket \tau \rrbracket_{\mathcal{U}}$ , and so on. Up to transposition, then, the interpretation of  $T$  generated from this interpretation of the basic types and terms exactly duplicates the construction of the free cartesian closed category over the basic types and terms, namely  $\mathbf{C}_T$ . This means, in particular, that since every arrow  $f : 1 \rightarrow \tau^{\sigma}$  in  $\mathbf{C}_T$  corresponds to a unique arrow  $\tilde{f} : \sigma \rightarrow \tau$ , since each  $\tilde{f} : \sigma \rightarrow \tau$  corresponds to a closed term  $M : \sigma \rightarrow \tau$ , and since each such term is interpreted as  $\llbracket M \rrbracket_{\mathcal{U}} = \tilde{f} = f : 1 \rightarrow \tau^{\sigma}$ , every arrow  $f : 1 \rightarrow \tau^{\sigma}$  is the interpretation under  $\mathcal{U}$  of a closed term of  $\mathcal{L}(T)$ . The same is true of  $f : 1 \rightarrow \sigma$ , where  $\sigma$  is not a function type. Hence  $\mathcal{U} : T \rightarrow \mathbf{C}_T$  is functionally complete.

To check that  $\mathcal{U}$  is a model of  $T$ , we need only verify that for all closed terms  $M$  and  $N$ ,  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$  if and only if  $T \vdash M = N$ . Clearly, whenever  $T \vdash M = N$  it must also be the case that  $M$  and  $N$  correspond to the same arrow in  $\mathbf{C}_T$ , by construction. Since transposition determines a bijection, then, it must be the case that  $M^* = N^*$ , which means that  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ . We may also show that the converse holds, i.e. that  $\mathcal{U}$  is complete: if  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ , then  $M^* = N^*$ , which implies that  $M = N$  as arrows in  $\mathbf{C}_T$ . By the definition of equality of arrows in  $\mathbf{C}_T$ , then,  $T \vdash M = N$ . Hence  $\mathcal{U}$  is a complete model of  $T$  in  $\mathbf{C}_T$  and moreover, in light of the previous paragraph, it is also functionally complete. This means, of course, that  $\mathcal{U}$  is a representation of  $T$  in  $\mathbf{C}_T$ .

We will now spend the remainder of this section examining the proof and the

consequences of the proposition below. Readers familiar with the abstract characterization of free constructions should note that it merely asserts that  $\mathbf{C}_T$  possesses the defining universal property of a free object.

**Proposition 2.2.1** *Let  $T$  be a typed  $\lambda$ -theory. For any standard model  $\mathcal{M}$  of  $T$  in a cartesian closed category  $\mathbf{C}$ , there is a unique cartesian closed functor  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$  such that the following (schematic) diagram commutes:*

$$\begin{array}{ccc}
 & & \mathbf{C} \\
 & \xrightarrow{\mathcal{M}^*} & \\
 \mathbf{C}_T & \xrightarrow{\quad \quad \quad} & \\
 \uparrow \mathcal{U} & \nearrow \mathcal{M} & \\
 T & & 
 \end{array}$$

**Proof** The construction of  $\mathcal{M}^*$  is clear: For any object of  $\mathbf{C}_T$ , say  $\llbracket \sigma \rrbracket_{\mathcal{U}}$ , define  $\mathcal{M}^*(\llbracket \sigma \rrbracket_{\mathcal{U}}) = \llbracket \sigma \rrbracket_{\mathcal{M}}$ . Since  $\mathcal{M}$  is standard, it interprets the terminal type 1, product types  $\sigma \times \tau$ , and function types  $\sigma \rightarrow \tau$  in a reasonable way:  $\llbracket 1 \rrbracket_{\mathcal{M}} = 1$ ,  $\llbracket \sigma \times \tau \rrbracket_{\mathcal{M}} = \llbracket \sigma \rrbracket_{\mathcal{M}} \times \llbracket \tau \rrbracket_{\mathcal{M}}$ , and  $\llbracket \tau^\sigma \rrbracket_{\mathcal{M}} = \llbracket \tau \rrbracket_{\mathcal{M}}^{\llbracket \sigma \rrbracket_{\mathcal{M}}}$ . Given our definition of the object function of  $\mathcal{M}^*$ , this means that

$$\mathcal{M}^*(1) = \mathcal{M}^*(\llbracket 1 \rrbracket_{\mathcal{U}}) = \llbracket 1 \rrbracket_{\mathcal{M}} = 1,$$

$$\begin{aligned}
 \mathcal{M}^*(\llbracket \sigma \rrbracket_{\mathcal{U}} \times \llbracket \tau \rrbracket_{\mathcal{U}}) &= \mathcal{M}^*(\llbracket \sigma \times \tau \rrbracket_{\mathcal{U}}) = \llbracket \sigma \times \tau \rrbracket_{\mathcal{M}} = \llbracket \sigma \rrbracket_{\mathcal{M}} \times \llbracket \tau \rrbracket_{\mathcal{M}} \\
 &= \mathcal{M}^*(\llbracket \sigma \rrbracket_{\mathcal{U}}) \times \mathcal{M}^*(\llbracket \tau \rrbracket_{\mathcal{U}})
 \end{aligned}$$

$$\begin{aligned}
 \mathcal{M}^*(\llbracket \tau \rrbracket_{\mathcal{U}}^{\llbracket \sigma \rrbracket_{\mathcal{U}}} \rrbracket_{\mathcal{U}}) &= \mathcal{M}^*(\llbracket \tau^\sigma \rrbracket_{\mathcal{U}}) = \llbracket \tau^\sigma \rrbracket_{\mathcal{M}} = \llbracket \tau \rrbracket_{\mathcal{M}}^{\llbracket \sigma \rrbracket_{\mathcal{M}}} \\
 &= \mathcal{M}^*(\llbracket \tau \rrbracket_{\mathcal{U}})^{\mathcal{M}^*(\llbracket \sigma \rrbracket_{\mathcal{U}})}
 \end{aligned}$$

Thus, at least as far as its object function is concerned,  $\mathcal{M}^*$  behaves like a cartesian closed functor.

For any arrow in  $\mathbf{C}_T$ , say  $\llbracket \Gamma \vdash M : \tau \rrbracket_{\mathcal{U}}$ , define  $\mathcal{M}^*(\llbracket \Gamma \vdash M : \tau \rrbracket_{\mathcal{U}}) = \llbracket \Gamma \vdash M : \tau \rrbracket_{\mathcal{M}}$ . Given this definition, one can easily check that  $\mathcal{M}^*$  satisfies the remaining equations that are required to hold if it is to be cartesian closed. For example, the maps  $!_{\llbracket \sigma \rrbracket_{\mathcal{U}}} : \llbracket \sigma \rrbracket_{\mathcal{U}} \rightarrow \llbracket 1 \rrbracket_{\mathcal{U}}$  in  $\mathbf{C}_T$  are the interpretations of terms in context of the form  $x : \sigma \vdash * : 1$ . Thus

$$\begin{aligned}
 \mathcal{M}^*(!_{\llbracket \sigma \rrbracket_{\mathcal{U}}}) &= \mathcal{M}^*(\llbracket x : \sigma \vdash * : 1 \rrbracket_{\mathcal{U}}) = \llbracket x : \sigma \vdash * : 1 \rrbracket_{\mathcal{M}} = !_{\llbracket \sigma \rrbracket_{\mathcal{M}}} \\
 &= !_{\mathcal{M}^*(\llbracket \sigma \rrbracket_{\mathcal{U}})},
 \end{aligned}$$

as required. Or, for  $f : \llbracket \sigma \rrbracket_{\mathcal{U}} \rightarrow \llbracket \tau_1 \rrbracket_{\mathcal{U}}$  and  $g : \llbracket \sigma \rrbracket_{\mathcal{U}} \rightarrow \llbracket \tau_2 \rrbracket_{\mathcal{U}}$ ,

$$\begin{aligned}
\mathcal{M}^*(f, g) &= \mathcal{M}^*(\llbracket x : \sigma \vdash M : \tau_1 \rrbracket_{\mathcal{U}}, \llbracket x : \sigma \vdash N : \tau_2 \rrbracket_{\mathcal{U}}) \\
&= \mathcal{M}^*(\llbracket x : \sigma \vdash \langle M, N \rangle : \tau_1 \times \tau_2 \rrbracket_{\mathcal{U}}) \\
&= \llbracket x : \sigma \vdash \langle M, N \rangle : \tau_1 \times \tau_2 \rrbracket_{\mathcal{M}} \\
&= (\llbracket x : \sigma \vdash M : \tau_1 \rrbracket_{\mathcal{M}}, \llbracket x : \sigma \vdash N : \tau_2 \rrbracket_{\mathcal{M}}) \\
&= (\mathcal{M}^*(\llbracket x : \sigma \vdash M : \tau_1 \rrbracket_{\mathcal{U}}), \mathcal{M}^*(\llbracket x : \sigma \vdash N : \tau_2 \rrbracket_{\mathcal{U}}))
\end{aligned}$$

By a few additional calculations of this kind, one can show that  $\mathcal{M}^*$  is indeed cartesian closed.  $\square$

As an immediate application of this result and a demonstration of its utility, we prove the following:

**Proposition 2.2.2** *The system of semantics consisting of all models in cartesian closed categories is complete and strongly complete.*

**Proof:** We have already shown that every  $\lambda$ -theory  $T$  has a representation  $\mathcal{U}$  in its syntactic category  $\mathbf{C}_T$ , meaning that cartesian closed semantics is strongly complete. To prove that cartesian closed semantics is complete, we must show that for any theory  $T$  and terms  $M$  and  $N$ ,  $T \vdash M = N$  if and only if  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$  for every model  $\mathcal{M}$  of  $T$  in a cartesian closed category. Clearly, if  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$  for all such models, then it must be true in the case of the model  $\mathcal{U}$ :  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ . This means, of course, that  $T \vdash M = N$ . Conversely, if  $T \vdash M = N$ , then  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ . By Proposition 2.2.1, every model  $\mathcal{M}$  of  $T$  factors as  $\mathcal{M}^*(\mathcal{U})$  for some cartesian closed functor  $\mathcal{M}^*$ . It follows that whenever  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ ,  $\llbracket M \rrbracket_{\mathcal{M}} = \mathcal{M}^*\llbracket M \rrbracket_{\mathcal{U}} = \mathcal{M}^*\llbracket N \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{M}}$  for any such  $\mathcal{M}$ .  $\square$

A more significant consequence of Proposition 2.2.1, perhaps, is that it allows us to relate the completeness of models in arbitrary cartesian closed categories to straightforward properties of functors in **Cart**, the most important of which we will now review. To begin, a functor  $F$  between categories  $\mathbf{C}$  and  $\mathbf{D}$  is said to be full (resp. faithful) if for every pair of objects  $A$  and  $B$  of  $\mathbf{C}$ ,  $F$  induces a surjection (resp. injection) between  $\text{Hom}_{\mathbf{C}}(A, B)$  and  $\text{Hom}_{\mathbf{D}}(F(A), F(B))$ . In other words,  $F : \mathbf{C} \rightarrow \mathbf{D}$  is full if for every arrow  $g : F(A) \rightarrow F(B)$  in  $\mathbf{D}$  there exists an  $f : A \rightarrow B$  in  $\mathbf{C}$  such that  $F(f) = g$ . On the other hand,  $F : \mathbf{C} \rightarrow \mathbf{D}$  is faithful if for any arrows  $f$  and  $g$  in  $\mathbf{C}$ ,  $F(f) = F(g)$  if and only if  $f = g$ . Also, a collection of

functors  $F : \mathbf{C} \rightarrow \mathbf{D}$  are said to be jointly faithful if for any arrows  $f$  and  $g$  in  $\mathbf{C}$ ,  $f = g$  if and only if  $F(f) = F(g)$  for every functor  $F$  in the collection.

Now, let  $\mathbf{C}$  be an arbitrary cartesian closed category, and let  $T$  be a typed  $\lambda$ -theory. First, one can show that if the collection of all cartesian closed functors  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$  are jointly faithful, then  $\mathbf{C}$ -valued semantics is complete for  $T$ . But this is clear: for any terms  $M$  and  $N$  of  $T$ ,  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$  for all models  $\mathcal{M} : T \rightarrow \mathbf{C}$  if and only if  $\mathcal{M}^* \llbracket M \rrbracket_{\mathcal{U}} = \mathcal{M}^* \llbracket N \rrbracket_{\mathcal{U}}$  for all cartesian closed functors  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$ . Since the  $\mathcal{M}^*$  are jointly faithful, this equation holds if and only if  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ , which in turn is the case if and only if  $T \vdash M = N$ . Thus  $\mathbf{C}$ -valued semantics are in fact complete.

Now let  $\mathcal{M} : T \rightarrow \mathbf{C}$  be an arbitrary model of  $T$  in  $\mathbf{C}$ . Suppose that the corresponding functor  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$  is full. Then for any types  $\sigma$  and  $\tau$  of  $T$ , any arrow  $g$  in  $\mathbf{C}$  between  $\llbracket \sigma \rrbracket_{\mathcal{M}} = \mathcal{M}^* \llbracket \sigma \rrbracket_{\mathcal{U}}$  and  $\llbracket \tau \rrbracket_{\mathcal{M}} = \mathcal{M}^* \llbracket \tau \rrbracket_{\mathcal{U}}$  is the image under  $\mathcal{M}^*$  of an arrow  $f : \llbracket \sigma \rrbracket_{\mathcal{U}} \rightarrow \llbracket \tau \rrbracket_{\mathcal{U}}$ . Since  $\mathcal{U}$  is a representation,  $f = \llbracket M \rrbracket_{\mathcal{U}}$  for some term  $M$  in  $T$ . Hence  $g = \llbracket M \rrbracket_{\mathcal{M}}$  for some  $M$ , meaning that  $\mathcal{M}$  is functionally complete.

If  $\mathcal{M}^*$  is faithful, I claim that the model  $\mathcal{M} : T \rightarrow \mathbf{C}$  is complete. To see that this is the case, one need only observe that if  $\mathcal{M}^*$  is indeed faithful, then for any terms  $M$  and  $N$ ,  $\mathcal{M}^* \llbracket M \rrbracket_{\mathcal{U}} = \mathcal{M}^* \llbracket N \rrbracket_{\mathcal{U}}$  (i.e.  $\llbracket M \rrbracket_{\mathcal{M}} = \llbracket N \rrbracket_{\mathcal{M}}$ ) if and only if  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$ . By definition, though,  $\llbracket M \rrbracket_{\mathcal{U}} = \llbracket N \rrbracket_{\mathcal{U}}$  if and only if  $T \vdash M = N$ . Hence  $\mathcal{M}^* \llbracket M \rrbracket_{\mathcal{U}} = \mathcal{M}^* \llbracket N \rrbracket_{\mathcal{U}}$  if and only if  $T \vdash M = N$  and, by definition,  $\mathcal{M} : T \rightarrow \mathbf{C}$  is complete.

Clearly, then, if  $\mathcal{M}^* : \mathbf{C}_T \rightarrow \mathbf{C}$  is both full and faithful, the corresponding model  $\mathcal{M} : T \rightarrow \mathbf{C}$  is a representation.

## 2.3 Presheaf Semantics

We may now apply the result of the previous section to prove that every  $\lambda$ -theory has a representation in the category  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  of presheaves on its syntactic category  $\mathbf{C}_T$ . First, though, it seems appropriate to briefly review the structure of presheaf categories in general. Our basic source is [11].

Let  $\mathbf{C}$  be an arbitrary small category, i.e. a category such that the collections of objects and arrows are sets in some suitable mathematical universe. The category of presheaves on  $\mathbf{C}$  has as its objects all contravariant functors from  $\mathbf{C}$  to  $\mathbf{Set}$  or, in simpler terms, all functors that assign to each object  $A$  of  $\mathbf{C}$  a set  $P(A)$  and assign to each  $\mathbf{C}$ -arrow  $f : A \rightarrow B$  a set mapping  $P(f) : P(B) \rightarrow P(A)$ . The arrows

between objects  $P$  and  $Q$  in  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  are all natural transformations  $\mu : P \rightarrow Q$  each of which consists of a family of set mappings  $\mu_A : P(A) \rightarrow Q(A)$ , indexed over the objects of  $\mathbf{C}$ , such that for every  $f : B \rightarrow A$ , the following diagram commutes:

$$\begin{array}{ccc} P(A) & \xrightarrow{\mu_A} & Q(A) \\ P(f) \downarrow & & \downarrow Q(f) \\ P(B) & \xrightarrow{\mu_B} & Q(B) \end{array}$$

For any object  $A$  of  $\mathbf{C}$ , there is a presheaf  $y(A) = \text{Hom}_{\mathbf{C}}(-, A)$ , the contravariant Hom functor, which is defined on objects  $B$  of  $\mathbf{C}$  by the equation

$$y(A)(B) = \text{Hom}_{\mathbf{C}}(B, A),$$

where  $\text{Hom}_{\mathbf{C}}(B, A)$  denotes the set of arrows from  $B$  to  $A$ . For any arrow  $f : B \rightarrow C$ , we define

$$y(A)(f) = \text{Hom}_{\mathbf{C}}(f, A) : \text{Hom}_{\mathbf{C}}(C, A) \rightarrow \text{Hom}_{\mathbf{C}}(B, A),$$

the function that sends  $g : C \rightarrow A$  in  $\text{Hom}_{\mathbf{C}}(C, A) = y(A)(C)$  to  $g \circ f : B \rightarrow A$  in  $\text{Hom}_{\mathbf{C}}(B, A) = y(A)(B)$ . Clearly, then, the map  $y$  assigns a presheaf to each object of  $\mathbf{C}$ . In fact,  $y$  extends to a covariant functor from  $\mathbf{C}$  to  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  if we define that for any  $f : B \rightarrow C$ ,  $y(f)$  is the natural transformation from  $\text{Hom}_{\mathbf{C}}(-, B)$  to  $\text{Hom}_{\mathbf{C}}(-, C)$ , i.e. from  $y(B)$  to  $y(C)$ , whose components are set mappings

$$y(f)_D : \text{Hom}_{\mathbf{C}}(D, B) \rightarrow \text{Hom}_{\mathbf{C}}(D, C)$$

given by  $y(f)_D(g) = f \circ g$  for all  $g : D \rightarrow B$ . Clearly, the map  $y$  thus defined is functorial from  $\mathbf{C}$  to  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ .

Furthermore, for the record,  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is cartesian closed. The terminal object of  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is (up to isomorphism) the functor  $1 : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$  that assigns to each object  $A$  a set  $1(A) = \{*_A\}$ , where  $\{*_A\}$  is some singleton. It should be clear that for any presheaf  $P$  there exists a unique natural transformation  $!_P : P \rightarrow 1$ , namely the one whose components are set maps  $(!_P)_A : P(A) \rightarrow \{*_A\}$  that send each  $x \in P(A)$  to  $*_A$ . Given any  $P$  and  $Q$  in  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ , the product  $P \times Q$  is computed “pointwise.” In other words,  $P \times Q$  is defined to be the map that assigns to each object  $A$  of  $\mathbf{C}$  the cartesian product  $P(A) \times Q(A)$  in the category of sets. The projections  $p_1 : P \times Q \rightarrow P$  and  $p_2 : P \times Q \rightarrow Q$  are natural transformations, with components

$p_{1,A} : P(A) \times Q(A) \rightarrow P(A)$  and  $p_{2,A} : P(A) \times Q(A) \rightarrow Q(A)$  corresponding to the obvious projections in **Set**. In addition, for any  $f : A \rightarrow B$  in **C** we define  $(P \times Q)(f)$  to be the arrow from  $P(B) \times Q(B)$  to  $P(A) \times Q(A)$  that makes the following diagram commute:

$$\begin{array}{ccccc}
 P(B) & \xleftarrow{p_{1,B}} & P(B) \times Q(B) & \xrightarrow{p_{2,B}} & Q(B) \\
 \downarrow P(f) & & \vdots (P \times Q)(f) & & \downarrow Q(f) \\
 P(A) & \xleftarrow{p_{1,A}} & P(A) \times Q(A) & \xrightarrow{p_{2,A}} & Q(A)
 \end{array}$$

The existence and uniqueness of this arrow follow from the fact that  $P(A) \times Q(A)$  is a product object in **Set**. The definition of the exponential of  $P$  and  $Q$  is more complicated, but still manageable. For every  $A$  in **C**, we define

$$Q^P(A) = \text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(A) \times P, Q),$$

the set of natural transformations between the presheaves  $y(A) \times P$  and  $Q$ . For any  $f : B \rightarrow A$ , we define  $Q^P(f)$  to be the map from  $\text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(A) \times P, Q)$  to  $\text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(B) \times P, Q)$  given by precomposition with the natural transformation  $(y(f) \circ p_1, p_2) : y(B) \times P \rightarrow y(A) \times P$ . The evaluation arrow  $\epsilon : Q^P \times P \rightarrow Q$  is defined to be the natural transformation whose components are  $\epsilon_A$  such that for any  $\theta \in \text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(A) \times P, Q)$  and  $x \in P(A)$ ,  $\epsilon_A(\theta, x) = \theta_A(1_A, x)$ , where  $\theta_A$  denotes the component of  $\theta$  at  $A$ . From these definitions, the reader can easily show that  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is cartesian closed. We now state the key result of this section:

**Proposition 2.3.1** *For any small category  $\mathbf{C}$ , the Yoneda embedding*

*$y : \mathbf{C} \rightarrow \mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is cartesian closed. Moreover, it is full, faithful, and injective on objects.*

**Proof:** Since the proof that  $y$  is cartesian closed is likely to be utterly transparent to readers who have had prior experience with the formalism and utterly opaque to those who have not, only a brief sketch will be provided here. Given a product object  $A \times B$  in **C**,  $y(A \times B) = \text{Hom}_{\mathbf{C}}(-, A \times B)$ , which (as we saw above) is isomorphic to  $\text{Hom}_{\mathbf{C}}(-, A) \times \text{Hom}_{\mathbf{C}}(-, B)$ , i.e. to  $y(A) \times y(B)$ . Given  $A$  and  $B$  in

$\mathbf{C}$ , similar considerations allow us to infer that

$$\begin{aligned}
y(B)^{y(A)}(C) &= \text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(C) \times y(A), y(B)) \\
&\cong \text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(C \times A), y(B)) \\
&= \text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(\text{Hom}_{\mathbf{C}}(-, C \times A), \text{Hom}_{\mathbf{C}}(-, B)) \\
&\cong \text{Hom}_{\mathbf{C}}(C \times A, B) \\
&\cong \text{Hom}_{\mathbf{C}}(C, B^A) \\
&= y(B^A)(C)
\end{aligned}$$

One can also check that  $y(B^A)$  and  $y(B)^{y(A)}$  agree on the arrows of  $\mathbf{C}$ , thereby completing the proof that  $y(B^A) = y(B)^{y(A)}$ . The verification that  $y$  has all the other necessary properties is left as an exercise.

To prove that  $y$  is full and faithful, we require the Yoneda Lemma, one of the most powerful and commonly used results in category theory:

**Lemma 2.3.1 (Yoneda)** *Let  $\mathbf{C}$  be a small category, and let  $A$  be an object of  $\mathbf{C}$ . If  $P$  is an object of  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ , then there is a bijection*

$$\text{Hom}_{\mathbf{Set}^{\mathbf{C}^{\text{op}}}}(y(A), P) \cong P(A).$$

While proofs of the lemma can be found in most reputable category theory texts, they involve a forbidding amount of detail. As a result, no such proof will be presented here. (Readers who are not inclined to take this result on faith may wish to consult [10].) At any rate, it is immediate from the lemma that given any objects  $A$  and  $B$  in  $\mathbf{C}_T$ , there is a bijection

$$\text{Hom}_{\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}}(y(A), y(B)) \cong y(B)(A) = \text{Hom}_{\mathbf{C}}(A, B).$$

Moreover, as one can check, this bijection is induced by  $y$ . Hence  $y$  is full and faithful.  $\square$

It is high time that we returned to our particular case, involving  $\mathbf{C}_T$  and  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$ , and addressed the ramifications of the general results outlined above. But this is a simple matter. Given any  $\lambda$ -theory  $T$ , it is apparent from the construction that  $\mathbf{C}_T$  is a small category. Hence  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is a cartesian closed category with products and exponentials as described above. More importantly, the Yoneda embedding of  $\mathbf{C}_T$  in  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is full and faithful. In light of the discussion in Section 2.2, then, we may infer that  $y \circ \mathcal{U} : T \rightarrow \mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is a representation. While this result is useful

as an illustration of the general approach that we will henceforth employ, in which we analyze semantics in arbitrary cartesian closed categories in terms of functors from the syntactic category, it also serves as the foundation of the proof of the sheaf representation theorem, as we will soon see.

# Chapter 3

## The Sheaf Representation

Using the basic facts about the semantics of the  $\lambda$ -calculus that were laid out in the previous chapter, we are now in a position to prove highly nontrivial results concerning the completeness of various systems of semantics. Before we may proceed to establish the fibration representation theorem, however, we require several intermediate completeness results, which are to be the subject matter of this chapter. In particular, we here consider the sheaf representation theorem of [1], which states that every  $\lambda$ -theory  $T$  has a representation in a category  $\mathbf{Sh}(X_T)$  of sheaves on a topological space  $X_T$ . We also prove an easy corollary, the improved presheaf representation mentioned in the introduction to Chapter 2. Practically speaking, Section 1 provides an elementary discussion of the notion of a sheaf on a topological space  $X$  and, in addition, of the category of sheaves on  $X$ . Readers already familiar with this material may wish to skip ahead to Section 2, which includes a precise statement of Awodey's sheaf representation theorem, as well as a sketch of its proof. In Section 3, we make use of the fact that the category of sheaves on a topological space is a full subcategory of the category of presheaves, inferring from Awodey's result that every  $\lambda$ -theory  $T$  has a representation in a category of the form  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ .

### 3.1 Sheaves

The field of algebraic topology, which has witnessed such dramatic advances in recent years, is based on the insight that fundamentally topological questions concerning the structure and classification of topological spaces may be translated into much simpler algebraic problems through the judicious assignment of algebraic structures (such as groups, rings, or algebras) either to the spaces at hand or to the open subsets

thereof. As one may clearly see, this insight has a distinctly categorical, functorial flavor. Indeed, as it turns out, all of the constructions of this kind that are studied in algebraic topology are functorial. (Nor should this come as a surprise, since it was considerations of this kind that led Eilenberg and Mac Lane to develop the foundations of category theory and, in particular, the notion of a functor.) One of the essential tools with which the assignment of algebraic structure may be effected is the categorical gadget known as a sheaf, a contravariant functor defined on the poset of open subsets of a topological space which satisfies a certain “collation” condition that will be described in detail below. While one is generally interested in sheaves of groups, rings, and other structured sets (that is, sheaves whose codomains are the categories of groups, rings, and so forth), we will be concerned here with the more general case of **Set**-valued sheaves.

Indeed, it is only natural that our concerns should diverge from those of the algebraic topologist, as we are here preoccupied with a dramatically different aspect of sheaf theory, namely its applications in logic and, in particular, in model theory. As an example of the value of the sheaf-theoretic perspective, one need only notice that a sheaf of groups on a topological space, say, may be regarded as a continuously varying model of the first order theory of groups, a generalization of the usual “constant” models in **Set**. Since the sheaf models of algebraic structures are more general, we might reasonably expect them to have properties over and above those possessed by constant ones. Thus, for example, it is possible to construct a sheaf model of the real numbers  $\mathbf{R}$  on which all functions  $f : \mathbf{R} \rightarrow \mathbf{R}$  are continuous. Similarly, the sheaves of sets that we consider here possess novel properties distinct from those of constant sets. In particular, we will soon see that unlike **Set**, the categories of **Set**-valued sheaves constitute strongly complete semantics for the  $\lambda$ -calculus.

Before we introduce the formal definition of sheaves and categories of sheaves, we would do well to consider the standard motivating example: the sheaf of continuous functions. To that end, let  $X$  be a topological space. Furthermore, let  $\mathcal{O}(X)$  be the set of open subsets of  $X$ , partially ordered by the relation of subset inclusion. Notice that we may also envision  $\mathcal{O}(X)$  as a category whose objects are the open subsets of  $X$  and whose morphisms correspond precisely with the order relation on  $\mathcal{O}(X)$ . In other words, there is a morphism between objects  $U$  and  $V$  in  $\mathcal{O}(X)$  if and only if  $U \subseteq V$ , in which case the morphism is the obvious inclusion mapping  $i_{UV} : U \hookrightarrow V$ .

Now, let  $C$  be the map that assigns to each  $U$  in  $\mathcal{O}(X)$  the set of continuous real-valued functions defined on  $U$ :

$$C(U) = \{f : U \rightarrow \mathbf{R} \mid f \text{ continuous}\}.$$

It will no doubt come as no surprise to the reader that given any continuous function  $f$  defined on an object  $U$  in  $\mathcal{O}(X)$ , if  $V \subseteq U$ , then the restriction of  $f$  to  $V$ , denoted  $f|_V$ , is a continuous function on  $V$ . In other words, whenever  $V \subseteq U$ , there is a restriction map  $|_V : C(U) \rightarrow C(V)$ . Moreover, the operation of restriction is transitive: if  $W \subseteq V \subseteq U$ , then  $(f|_V)|_W = f|_W$  for any  $f \in C(U)$ . This means, of course, that the restriction map  $|_W : C(U) \rightarrow C(W)$  corresponding to the “composition”  $W \subseteq U$  of the inclusions  $W \subseteq V \subseteq U$ , is equal to the composition of the restrictions  $|_V : C(U) \rightarrow C(V)$  and  $|_W : C(V) \rightarrow C(W)$ , meaning that  $C$  is functorial from  $\mathcal{O}(X)$  to  $\mathbf{Set}$ . Moreover, since  $C$  is contravariant, it must be a presheaf on  $\mathcal{O}(X)$ .

Recalling that the continuous functions on a topological space have a number of additional, less trivial properties, one can see that  $C$  is not merely a presheaf, but rather possesses considerably more structure. In particular, we recall the “collation” or “matching” property. Let  $U$  be an open subset of  $X$ , and let  $U = \bigcup_{i \in I} U_i$  be an open cover of  $U$ , where  $I$  is some index set. Then for any  $I$ -indexed family of mappings  $\{f_i\}$ ,  $f_i \in C(U_i)$ , if every  $f_i$  and  $f_j$  agree on the intersection of their domains ( $f_i|_{U_i \cap U_j} = f_j|_{U_i \cap U_j}$  for all  $i$  and  $j$  in  $I$ ), there is a unique  $f \in C(U)$  such that  $f|_{U_i} = f_i$  for all  $i$ . More formally, we regard the family  $\{f_i\}$  as an element of the product  $\prod_{i \in I} C(U_i)$ , and the doubly-indexed family  $\{f_i|_{U_i \cap U_j}\}$  as an element of  $\prod_{i,j \in I} C(U_i \cap U_j)$ . Let  $e : C(U) \rightarrow \prod_{i \in I} C(U_i)$  be the map that takes  $f$  to  $\{f|_{U_i}\}$ , and let  $p, q : \prod_{i \in I} C(U_i) \rightarrow \prod_{i,j \in I} C(U_i \cap U_j)$  be the maps that take  $\{f_i\}$  to  $\{f_i|_{U_i \cap U_j}\}$  and  $\{f_j|_{U_i \cap U_j}\}$ , respectively. It is clear that  $p$  and  $q$  must agree on  $\{f_i\}$  if it is in the image of  $e$  and, in light of the property described above, it is also the case that they agree on  $\{f_i\}$  only if it lies in the image of  $e$ .

In fact, a sheaf on the space  $X$  will be defined as a presheaf on  $\mathcal{O}(X)$  that satisfies precisely this condition. First, though, we must rephrase it in terms applicable to presheaves on  $\mathcal{O}(X)$  in general. To that end, we consider not  $C$ , but rather an arbitrary presheaf  $F$ , and recall that whenever  $V \subseteq U$  in  $\mathcal{O}(X)$ , the image of this inclusion under  $F$  is a map  $F(V \subseteq U) : F(U) \rightarrow F(V)$ . Since  $F$  need not be a sheaf of function sets, this map will not necessarily have the same natural interpretation that it did in the case of  $C$ . Nevertheless, we may still regard it as a

kind of generalized restriction map, and continue to denote it by  $|_V$  when the relevant presheaf is clear from the context. With respect to this new notion of restriction, given any open cover  $U = \bigcup_{i \in I} U_i$  we again define  $e : F(U) \rightarrow \prod_{i \in I} F(U_i)$  to be the map given by  $e(t) = t|_{U_i}$ , and for any collection  $\{t_i\} \in \prod_{i \in I} F(U_i)$  we define the maps  $p, q : \prod_{i \in I} F(U_i) \rightarrow \prod_{i, j \in I} F(U_i \cap U_j)$  by

$$p(\{t_i\}) = \{t_i|_{U_i \cap U_j}\} \quad \text{and} \quad q(\{t_i\}) = \{t_j|_{U_i \cap U_j}\}$$

We may now define the general notion of a sheaf:

**Definition 3.1.1** *A sheaf on a topological space  $X$  is a functor  $F : \mathcal{O}(X)^{\text{op}} \rightarrow \mathbf{Set}$  such that for every open cover  $U = \bigcup_{i \in I} U_i$  of an open set  $U$  in  $\mathcal{O}(X)$ ,  $F$  has the following property: for every  $\{t_i\} \in \prod_{i \in I} F(U_i)$ ,  $p(\{t_i\}) = q(\{t_i\})$  if and only if  $\{t_i\} = e(t)$  for some unique  $t \in F(U)$ .*

Now that we have a precise characterization of individual sheaves on a topological space  $X$ , we may form the category of sheaves on  $X$ ,  $\mathbf{Sh}(X)$ . Specifically,  $\mathbf{Sh}(X)$  is defined to be the category that has as its objects all sheaves on  $X$ , and has as its morphisms all natural transformations between them. Since every sheaf is a presheaf, it is a simple matter to convince oneself that  $\mathbf{Sh}(X)$  is a cartesian closed category, with products and exponentials identical to the ones in the presheaf categories considered in Section 2.3. Just to refresh our memories, the product of any sheaves  $F$  and  $G$  in  $\mathbf{Sh}(X)$  will be computed pointwise: for any  $U$  in  $\mathcal{O}(X)$ ,  $(F \times G)(U) = F(U) \times G(U)$ . As before, given any sheaves  $F$  and  $G$ , the exponential of  $G$  by  $F$  will be defined by

$$G^F(U) = \text{Hom}_{\mathbf{Sh}(X)}(y(U) \times F, G)$$

for all  $U$  in  $\mathcal{O}(X)$ , where  $y$  is the Yoneda embedding. Given  $U \subseteq V$  in  $\mathcal{O}(X)$ , the images under  $F \times G$  and  $G^F$  will be defined as in Section 2.3. Of course, one must check that the  $F \times G$  and  $G^F$  thus defined are sheaves and, moreover, that they are product and exponential objects in  $\mathbf{Sh}(X)$ , respectively. Readers who wish to see a proof of this fact are referred to [11].

Although there is a great deal more to be said on the subject of sheaves, we must now move on to other matters. Readers interested in further information on sheaf theory may wish to consult the authoritative source, [11].

## 3.2 The Representation Theorem

We may now state the central result of this chapter, the sheaf representation theorem of [1]. Unfortunately, due to the complexities involved in the proof, it is impossible to present it in the exhaustive detail to which the reader of this thesis may now be accustomed. Although an effort will be made to convey the essential ideas involved, the treatments given below should be regarded less as elements of an exhaustive demonstration than as gentle reminders for experienced readers and, for the rest, as invitations to further study. At any rate, we wish to prove the following result:

**Theorem 3.2.1** *For every typed  $\lambda$ -theory  $T$ , there exists a topological space  $X_T$  such that  $T$  has a representation in  $\mathbf{Sh}(X_T)$ .*

The proof of this theorem relies on two important results. First, for any  $\lambda$ -theory  $T$ , the category  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is not merely cartesian closed, but rather a topos, an even more highly structured category. Moreover, it has the property of having “enough points” (an idea that will be explained momentarily). Given that this is the case, the spatial covering theorem of [4], Theorem 3.2.2 below, allows us to conclude that there exists a fully faithful functor  $\phi : \mathbf{Set}^{\mathbf{C}_T^{\text{op}}} \rightarrow \mathbf{Sh}(X_T)$  for some topological space  $X_T$ , from which we infer the existence of a representation of  $T$  in  $\mathbf{Sh}(X_T)$ .

We begin by examining the notion of a topos, concerning ourselves only with the essential properties while, sadly, neglecting the deeper aspects of topoi and of topos theory (although readers who wish to explore such topics as the logical and foundational significance of topoi can find ample treatments in [5], [11], and [12]). In the following discussion, our general method will be to explain the defining structures of a topos in terms of the more familiar analogues occurring in the prototypical example of a topos, the category of sets. In particular, we define a topos to be a cartesian closed category with a subobject classifier, an object that may be best understood as a generalization of  $\Omega$ , the set of truth values in  $\mathbf{Set}$ .

To be more precise, in the context of  $\mathbf{Set}$ , we choose our  $\Omega$  to be the typical two element set  $\{0, 1\}$ , where 1 is interpreted as “true,” and 0 is interpreted as “false.” As the reader will no doubt recall,  $\Omega$  has the following property (relative to our choice of 1 as “true”): for any set  $S$  and  $R \subseteq S$ , there is a unique  $\chi^R : S \rightarrow \Omega$ , the characteristic function of  $R$  in  $S$ , defined by

$$\chi^R(s) = \begin{cases} 1 & s \in R \\ 0 & \text{otherwise} \end{cases}$$

Moreover,  $R$  is retrievable as  $(\chi^R)^{-1}(1) = \{s \in S \mid \chi^R(s) = 1\}$ .

To prepare ourselves for the generalization to arbitrary categories, it is in our interest to give a more formal description of this property. We begin by noticing that the terminal object of **Set** will be a singleton  $\{*\}$  (up to isomorphism), since any set  $S$  maps uniquely to  $\{*\}$  via the map  $!_S : S \rightarrow 1$  that sends every  $s \in S$  to  $*$ . Given that this is the case, we may define a function  $\text{true} : 1 \rightarrow \Omega$  that sends  $*$  to the element 1 of  $\Omega$ . Clearly this is consistent with our earlier choice of 1 as “true.” Since it is clearly the case that  $(\text{true} \circ !_R)(r) = \text{true}(*) = 1$  for all  $r \in R$ , and since, as we noted above,  $(\chi^R)^{-1}(1) = R$ , one can clearly see that  $R$  is the pullback along  $\text{true} : 1 \rightarrow \Omega$  of  $\chi^R$  or, equivalently, that the following diagram is a pullback

$$\begin{array}{ccc} R & \xrightarrow{!_R} & 1 \\ \downarrow r & & \downarrow \text{true} \\ S & \xrightarrow{\chi^R} & \Omega \end{array}$$

where  $r : R \hookrightarrow S$  is the inclusion of  $R$  in  $S$ . For those unfamiliar with the notion of a pullback, in this case it simply means that for any subset  $T \subseteq S$  and inclusion map  $t : T \rightarrow S$ ,  $(\chi^R \circ t)(x) = (\text{true} \circ !_T)(x) = 1$  for all  $x$  in  $T$  if and only if  $T \subseteq R$ , as one would expect. In fact, this is the characterization from which we will generalize to arbitrary categories: the set  $\Omega$  and the selected function  $\text{true} : 1 \rightarrow \Omega$  have the property that for any set  $S$  and any  $R \subseteq S$  there is a unique  $\chi^R : S \rightarrow \Omega$  such that the diagram above is a pullback.

In any category **C** there is an analogue of the relation of subset inclusion. In particular, just as a subset  $R$  of a given set  $S$  can be identified with the preferred injective map from  $R$  to  $S$ , namely the inclusion  $R \hookrightarrow S$ , a subobject of a given object  $B$  in **C** consists of an arrow  $a : A \rightarrow B$  that satisfies the categorical analogue of injectivity. To be precise, the arrow  $a : A \rightarrow B$  must be a monomorphism, defined by the condition that for any arrows  $x : C \rightarrow A$  and  $y : C \rightarrow A$ ,  $a \circ x = a \circ y$  if and only if  $x = y$ . In a slight abuse of language, we will often speak of the object  $A$  itself as a subobject of  $B$ , rather than the monomorphism  $a : A \rightarrow B$ . At any rate, we now translate our earlier description of the classifier  $\Omega, \text{true} : 1 \rightarrow \Omega$  to this more general context:

**Definition 3.2.1** *A subobject classifier in a category **C** is an object  $\Omega$  and an arrow  $\text{true} : 1 \rightarrow \Omega$  such that for any object  $B$  and subobject  $a : A \rightarrow B$ , there is a unique*

arrow  $\chi^a : B \rightarrow \Omega$  such that the following is a pullback diagram:

$$\begin{array}{ccc} A & \xrightarrow{!_A} & 1 \\ a \downarrow & & \downarrow \text{true} \\ B & \xrightarrow{\chi^a} & \Omega \end{array}$$

In some sense, this simply means that there is a unique  $\chi^a : B \rightarrow \Omega$  with the property that  $\chi^a \circ a = \text{true} \circ !_A$  and for any subobject  $c : C \rightarrow B$  of  $B$ ,  $\chi^a \circ c = \text{true} \circ !_C$  if and only if  $C$  is a subobject of  $A$ . We now return to the notion of a topos:

**Definition 3.2.2** *A topos is a cartesian closed category with a subobject classifier  $\Omega$ ,  $\text{true} : 1 \rightarrow \Omega$ .*

As suggested above, our task now is to prove that categories of the form  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  are topos. In fact, we prove a slightly stronger theorem, using [8] and [11] as guides:

**Proposition 3.2.1** *For any cartesian closed category  $\mathbf{C}$ ,  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is a topos.*

**Proof:** We already know that  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is cartesian closed, so it suffices merely to show that it contains a subobject classifier. In order to do so, however, we require the following additional notion:

**Definition 3.2.3** *A sieve on an object  $A$  in a category  $\mathbf{C}$  is a collection  $S$  of arrows with codomain  $A$  such that if  $f \in S$  and  $h$  is an arrow for which the composition  $f \circ h$  is defined, then  $f \circ h \in S$ .*

For an easy example of a sieve, one might consider the case in which  $\mathbf{C}$  is a poset. It should be clear that a sieve on an element  $p$  of  $\mathbf{C}$  will then be a collection  $S$  of elements of  $\mathbf{C}$  such that for any  $s \in S$ ,  $s \leq p$  and  $s' \in S$  for all  $s' \leq s$ , i.e. a lower set in  $\mathbf{C}$ . One should also notice that in any category  $\mathbf{C}$ , the collection of all arrows with codomain  $A$  is a sieve on  $A$ , which we henceforth denote by  $t(A)$ .

We may now construct the presheaf  $\Omega : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Set}$  that is to be our candidate for the role of subobject classifier. For any object  $A$ , we define

$$\Omega(A) = \{S \mid S \text{ is a sieve on } A \text{ in } \mathbf{C}\}$$

and for any  $f : A \rightarrow B$ , we define  $\Omega(f) : \Omega(B) \rightarrow \Omega(A)$  to be the map given by

$$\Omega(f)(S) = \{g \mid f \circ g \in S\}$$

for any sieve  $S$  on  $B$ . As a test of our understanding, we would do well to check that the set  $\Omega(f)(S)$  is indeed a sieve on  $A$ . This is easily accomplished. First, we note that any  $g \in S$  must have codomain  $A$  if it is to be composable with  $f$ . Furthermore, for any  $h$  such that  $g \circ h$  is defined, the fact that  $S$  is a sieve implies that  $f \circ g \in S$  and, similarly, that  $(f \circ g) \circ h \in S$ . In other words,  $f \circ (g \circ h) \in S$ , meaning that  $g \circ h \in \Omega(f)(S)$ . Sieves are also involved in the construction of the natural transformation  $\text{true} : 1 \rightarrow \Omega$ . Recalling that the terminal object of  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is a functor  $1$  that assigns to each object  $A$  a singleton  $\{*_A\}$ , we define  $\text{true}$  to be the natural transformation whose components  $\text{true}_A : 1(A) \rightarrow \Omega(A)$  map each  $*_A$  to the maximal sieve  $t(A)$  on  $A$ .

It remains to show that  $\Omega, \text{true} : 1 \rightarrow \Omega$  is in fact a subobject classifier. We recall, first, that in presheaf categories  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  a presheaf  $P$  is said to be a subobject (or, to be precise, a subfunctor) of a presheaf  $Q$  if and only if  $P(A) \subseteq Q(A)$  for all objects  $A$  and, moreover, the image of any  $f : B \rightarrow A$  under  $P$ ,  $P(f) : P(A) \rightarrow P(B)$ , is equal to the restriction of  $Q(f)$  to  $P(A)$ . Given any subobject  $P$  of  $Q$ , we define a classifying map  $\chi^P : Q \rightarrow \Omega$  whose component  $\chi_A^P : Q(A) \rightarrow \Omega(A)$  at each object  $A$  in  $\mathbf{C}$  is given by the following rule:

$$\chi_A^P(x) = \{f : B \rightarrow A \mid Q(f)(x) \in P(B)\}.$$

Let us check that this is in fact a sieve on  $A$ . To that end, suppose that  $f : B \rightarrow A$  is contained in  $\chi_A^P(x)$ , and let  $g$  be an arbitrary arrow with codomain  $B$ , say  $g : C \rightarrow B$ . By the familiar rule that contravariant functors reverse the order of composition, we have

$$\begin{aligned} Q(f \circ g)(x) &= (Q(g) \circ Q(f))(x) \\ &= Q(g)(Q(f)(x)) \end{aligned}$$

Since  $f$  is in  $\chi_A^P(x)$ ,  $Q(f)(x)$  must be an element of  $P(B)$ . Furthermore, by the definition of the subobject relation in  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ , the restriction of  $Q(g)$  to  $P(B)$  is simply  $P(g) : P(B) \rightarrow P(C)$ , meaning that  $Q(g)(Q(f)(x))$  is an element of  $P(C)$ . In other words,  $f \circ g : C \rightarrow A$  is contained in  $\chi_A^P(x)$ . Since this is true for every  $g$  such that  $f \circ g$  is defined,  $\chi_A^P(x)$  is a sieve on  $A$ .

Notice that if  $x \in P(A)$ , then every  $f : B \rightarrow A$  satisfies the condition that  $Q(f)(x) \in P(B)$ . Hence  $\chi_A^P(x) = t(A)$ . Conversely, one can easily see that if  $\chi_A^P(x) \neq t(A)$  or, equivalently, if there is some  $f : B \rightarrow A$  such that  $Q(f)(x) \notin P(B)$ , then  $x$  cannot be in  $P(A)$ . In other words,  $\chi_A^P(x) = t(A)$  if and only if  $x \in P(A)$ ,

as required. Now, let  $!_P$  denote the unique natural transformation from  $P$  to  $1$ , whose components  $(!_P)_A : P(A) \rightarrow 1(A)$  map each element of  $P(A)$  to  $*_A$ . Clearly,  $(\text{true}_A \circ (!_P)_A)(x) = t(A)$  for all  $x \in P(A)$ , meaning that  $\chi_A^P(x) = (\text{true}_A \circ (!_P)_A)(x)$  if and only if  $x \in P(A)$ . To put it another way, this means that the following diagram is a pullback in the category of sets:

$$\begin{array}{ccc} P(A) & \xrightarrow{(!_P)_A} & 1(A) \\ \downarrow & & \downarrow \text{true}_A \\ Q(A) & \xrightarrow{\chi_A^P} & \Omega(A) \end{array}$$

where the map  $P(A) \rightarrow Q(A)$  is the obvious inclusion. Now, since pullbacks in presheaf categories of the form  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  are computed componentwise, this is the same as saying that the following diagram is a pullback in  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ :

$$\begin{array}{ccc} P & \xrightarrow{!_P} & 1 \\ \downarrow & & \downarrow \text{true} \\ Q & \xrightarrow{\chi^P} & \Omega \end{array}$$

Moreover, one can show that  $\chi^P$  is the unique natural transformation such that this diagram is a pullback (see [11]), from which it follows that, by Definition 3.2.1,  $\Omega, \text{true} : 1 \rightarrow \Omega$  is indeed a subobject classifier (which is, incidentally, provably unique up to isomorphism). Hence  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  is a topos, as claimed.  $\square$

Since the syntactic category  $\mathbf{C}_T$  of any  $\lambda$ -theory  $T$  is cartesian closed, the proposition implies that  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is a topos, as desired.

For the sake of completeness, we must also give an explanation of the other property predicated of  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  in the opening lines of this proof, namely that of having “enough points.” It is difficult to motivate this idea in general, but in presheaf categories of the form  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ , where  $\mathbf{C}$  is any small cartesian closed category, it simply means that the collection of (finite limit- and colimit-preserving) functors from  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  to  $\mathbf{Set}$  is jointly faithful. In fact, as one can readily verify, it suffices to consider the family of evaluation functors  $\text{eval}_A : \mathbf{Set}^{\mathbf{C}^{\text{op}}} \rightarrow \mathbf{Set}$ , indexed by the objects  $A$  of  $\mathbf{C}$ , where  $\text{eval}_A$  assigns to each presheaf  $P$  the set  $P(A)$ , and

assigns to each natural transformation  $\mu : P \rightarrow Q$  the component of  $\mu$  at  $A$ ,  $\mu_A : P(A) \rightarrow Q(A)$ . Without going into too much detail, we remark that as a result of the componentwise definition of limits and colimits in  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$ , it is certainly the case that they will be preserved by the evaluation functors. Moreover, it is evident that the evaluation functors are jointly faithful as, given any natural transformations  $\mu$  and  $\eta$  between presheaves  $P$  and  $Q$ ,  $\text{eval}_A(\mu) = \text{eval}_A(\eta)$  for all  $\text{eval}_A$  if and only if  $\mu_A = \eta_A$  for all  $A$  in  $\mathbf{C}$ , in which case  $\mu = \eta$ . Thus the functors  $\text{eval}_A$  constitute a sufficient set of points for  $\mathbf{Set}^{\mathbf{C}^{\text{op}}}$  or, in our case, for  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$ .

This discussion, in conjunction with Proposition 3.2.1, suggests that  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is indeed a topos with enough points. Hence we may invoke the spatial covering theorem of [4]. For our purposes, we require only the following watered-down version:

**Theorem 3.2.2** *Let  $\mathcal{T}$  be a topos with enough points. Then there exists a topological space  $X$  such that there is a full, faithful, cartesian closed functor  $\phi$  from  $\mathcal{T}$  to  $\mathbf{Sh}(X)$ .*

Unfortunately, the proof of this theorem falls well beyond the scope of this humble enquiry. Readers interested in further details are advised to consult the original source.

At any rate, the theorem implies that for any  $\lambda$ -theory  $T$  there is a topological space  $X_T$  and a full and faithful cartesian closed functor  $\phi : \mathbf{Set}^{\mathbf{C}_T^{\text{op}}} \rightarrow \mathbf{Sh}(X_T)$ . Clearly, this is precisely what we need to prove the existence of a sheaf representation. Recalling that the Yoneda functor  $y : \mathbf{C}_T \rightarrow \mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is full, faithful, and cartesian closed, it should be clear that the composition  $\phi \circ y : \mathbf{C}_T \rightarrow \mathbf{Sh}(X_T)$  is also full, faithful, and cartesian closed. As we saw in Chapter 2, this means that  $\phi \circ y$  corresponds to a representation  $\llbracket - \rrbracket$  of  $T$  in  $\mathbf{Sh}(X_T)$ , given by the following commutative triangle:

$$\begin{array}{ccc}
 \mathbf{C}_T & \xrightarrow{\phi \circ y} & \mathbf{Sh}(X_T) \\
 \mathcal{U} \uparrow & \nearrow \llbracket - \rrbracket & \\
 T & & 
 \end{array}$$

where  $\mathcal{U}$  is the term model of  $T$  in its syntactic category,  $\mathbf{C}_T$ . More informally, one might simply note that the image of any representation under a full and faithful cartesian closed functor must also be a representation, from which it follows that  $\phi \circ (y \circ \mathcal{U})$  is a representation of  $T$  in  $\mathbf{Sh}(X_T)$ , as desired.

### 3.3 The Improved Presheaf Representation

We are now in a position to dramatically improve upon our earlier result concerning presheaf semantics, Proposition 2.3.1, which asserted that the class of models in categories of presheaves on cartesian closed categories constitutes strongly complete semantics. In particular, we may now prove that strong completeness will not be lost if we replace the class of categories of the form  $\mathbf{Set}^{C^{\text{op}}}$ , where  $C$  is arbitrary cartesian closed category, with the far smaller class of categories of the form  $\mathbf{Set}^{\mathcal{O}(X)^{\text{op}}}$ .

**Proposition 3.3.1** *Every typed  $\lambda$ -theory  $T$  has a representation in the presheaf category  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ .*

**Proof:** This is an easy consequence of Theorem 3.2.1. By definition, we know that the sheaves on  $X_T$  are presheaves on  $\mathcal{O}(X_T)$ , albeit with additional structure. Furthermore, the collection of morphisms between sheaves  $F$  and  $G$  in  $\mathbf{Sh}(X_T)$ ,  $\text{Hom}_{\mathbf{Sh}(X_T)}(F, G)$ , is identical to the set of all natural transformations between the presheaves underlying  $F$  and  $G$ , by definition. Now, let  $H$  be the forgetful functor from  $\mathbf{Sh}(X_T)$  to  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ , which assigns to each sheaf  $F$  the underlying presheaf  $H(F)$  and assigns to each sheaf morphism  $\eta : F \rightarrow G$  the underlying presheaf morphism  $H(\eta) : H(F) \rightarrow H(G)$ . It is clear from the above that  $H$  is injective on objects and, since

$$\text{Hom}_{\mathbf{Sh}(X_T)}(F, G) = \text{Hom}_{\mathbf{Set}^{C^{\text{op}}}}(H(F), H(G)),$$

$H$  is certainly both full and faithful. In addition, since the products and exponentials in  $\mathbf{Sh}(X_T)$  are defined in precisely the same way as those in  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ , it is also clear that  $H$  is cartesian closed.

From Theorem 3.2.1, we know that any  $\lambda$ -theory  $T$  has a representation in the category  $\mathbf{Sh}(X_T)$ . As before, the fact that  $H$  is full, faithful, and cartesian closed implies that the image of the sheaf representation will be a representation in  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ .  $\square$



# Chapter 4

## Fibration Semantics

Before we turn to the proof of the main result of this thesis, we would do well to review the arguments that have brought us to this point. Recall that we first observed that every lambda theory  $T$  has a representation in its syntactic category  $\mathbf{C}_T$ , namely the term model  $U$ . Using the fact that the image of a representation under a full and faithful cartesian closed functor is also a representation, we showed that the Yoneda embedding provides us with a representation of  $T$  in the presheaf category  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$ , namely  $y \circ U$ . Since  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  is a topos with “enough points,” the spatial covering theorem of [4] ensures that there is a fully faithful cartesian closed functor  $\phi$  from  $\mathbf{Set}^{\mathbf{C}_T^{\text{op}}}$  to the category  $\mathbf{Sh}(X_T)$  of sheaves on the topological space  $X_T$ . Hence  $\phi \circ y \circ U$  is a representation of  $T$  in  $\mathbf{Sh}(X_T)$ . Finally, since the forgetful functor  $H : \mathbf{Sh}(X_T) \rightarrow \mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$  is full, faithful, and cartesian closed, the image the sheaf representation under  $H$  is a representation of  $T$  in the category of presheaves on  $\mathcal{O}(X_T)$ , the poset of open sets of  $X_T$ .

In this chapter, we will see that this result leads us to a representation in  $\mathbf{Fib}(\mathcal{O}(X_T))$ , the category of fibrations over  $\mathcal{O}(X_T)$ . Section 1 is devoted to preparations for the proof of the main theorem and the description of the representation. In particular, we give the definition of a fibration over a poset, and discuss the nature of categories of poset fibrations. In addition, we provide a lengthy (and occasionally unpleasant) proof of an elegant and utterly indispensable technical result, the equivalence between the categories  $\mathbf{Fib}(P)$  and  $\mathbf{Set}^{P^{\text{op}}}$ , where  $P$  is any poset. We then use this equivalence to derive the cartesian closedness of  $\mathbf{Fib}(P)$ , and to analyze the products, exponentials, and operations of pairing and transposition. In Section 2, we note that the equivalence of categories implies the existence of a full and faithful cartesian closed functor from  $\mathbf{Set}^{P^{\text{op}}}$  to  $\mathbf{Fib}(P)$  or, more to the point,

from  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$  to  $\mathbf{Fib}(\mathcal{O}(X_T))$  in the special case in which  $P = \mathcal{O}(X_T)$ . In light of our improved presheaf representation theorem, then, it follows that every  $\lambda$ -theory has a representation in  $\mathbf{Fib}(\mathcal{O}(X_T))$ . We then conclude in the only natural way, with a brief discussion of this representation.

## 4.1 Poset Fibrations

At the risk of restating details with which the reader is already intimately acquainted, a brief discussion of posets, and of the category of posets, is now in order. Naturally, a poset  $P$  is a set equipped with a relation  $\leq$  that is reflexive, transitive, and anti-symmetric. As we often do with other kinds of structured sets, we may form the category of posets, denoted  $\mathbf{Pos}$ , that has as its objects all posets, and as its morphisms all structure-preserving maps between them. In this case, of course, the structure-preserving maps must be monotone functions, i.e. functions  $F : P \rightarrow Q$  such that  $F(p) \leq F(p')$  in  $Q$  whenever  $p \leq p'$  in  $P$ . One may also envision a poset  $P$  as a category whose objects are the elements of  $P$  and whose morphisms correspond precisely with the order relation on  $P$ . In particular, we say that for any objects  $p$  and  $p'$  of  $\mathbf{C}$ , there is a morphism from  $p$  to  $p'$  if and only if  $p \leq p'$ , in which case the morphism is unique and will be denoted by  $i_{pp'} : p \rightarrow p'$ . Indeed, this is precisely the interpretation that we adopted in our earlier description of the poset  $\mathcal{O}(X)$ , although in that case the arrows  $i_{UV} : U \rightarrow V$  had a concrete interpretation as inclusion mappings. In this view, which differs only formally from the previous one, the objects of  $\mathbf{Pos}$  are categories, and the morphisms between them are structure-preserving functors. One should note, though, that any functor between  $\mathbf{Pos}$  objects, regarded as categories, is monotone: if  $F : P \rightarrow Q$  is a functor,  $P$  and  $Q$  are posets, and  $p$  and  $p'$  are objects of  $P$ , then  $p \leq p'$  implies that there is a morphism  $i_{pp'} : p \rightarrow p'$ , which is sent to a morphism  $F(i_{pp'}) : F(p) \rightarrow F(p')$ , and the existence of such a morphism implies that  $F(p) \leq F(p')$ . Similarly, any monotone function may be interpreted as a functor.

The point of all this, of course, is that we may adopt whichever of the two formulations of the notions of posets and poset mappings that suits our purposes. Specifically, we will typically interpret them as structured sets and monotone functions when we are concerned with properties intrinsic to structures in  $\mathbf{Pos}$ . On the other hand, when we are interested in more general constructions on posets, such as the sheaves and presheaves considered elsewhere in this text, it is necessary to

consider the posets in their categorical guise. No doubt the reader will have no difficulty negotiating these subtleties.

One should note that the category **Pos** contains the familiar structures characteristic of a cartesian closed category. Given any posets  $P$  and  $Q$ , the product  $P \times Q$  is the set  $\{(p, q) \mid p \in P \text{ and } q \in Q\}$ , together with an order relation defined by the condition that  $(p, q) \leq (p', q')$  if and only if  $p \leq p'$  and  $q \leq q'$ . One can easily check that the canonical projections  $p_1 : P \times Q \rightarrow P$  and  $p_2 : P \times Q \rightarrow Q$  are monotone and, furthermore, that the pairing  $(f, g)$  of any two monotone functions  $f : R \rightarrow P$  and  $g : R \rightarrow Q$  is itself monotone. The exponential  $Q^P$  is the set of monotone functions  $f : P \rightarrow Q$ , with an ordering defined by the following condition: for any  $f$  and  $f'$  in  $Q^P$ ,  $f \leq f'$  if and only if  $f(p) \leq f'(p)$  for all  $p$  in  $P$ . Again, the arrows associated with this structure can be seen to be monotone.

We must now examine the nature of slice categories of the form **Pos**/ $P$ , where  $P$  is an arbitrary poset, as an understanding of such categories (and the objects thereof) is a precondition for any discussion of poset fibrations. In particular, given a poset  $P$ , **Pos**/ $P$  is defined to be the category whose objects are monotone functions with codomain  $P$ , i.e.  $e : E \rightarrow P$ . A morphism  $h$  between objects  $e : E \rightarrow P$  and  $f : F \rightarrow P$  is a monotone function  $h : E \rightarrow F$  such that the following diagram commutes:

$$\begin{array}{ccc} E & \xrightarrow{h} & F \\ & \searrow e & \swarrow f \\ & & P \end{array}$$

Since readers unfamiliar with the slice category construction will likely find this rather strange, we ought to go into a bit more detail. First, we notice that the identity arrow  $1_e : e \rightarrow e$  on an object  $e : E \rightarrow P$  of **Pos**/ $P$  will simply be the identity function on the poset  $E$ . Furthermore, given objects  $e : E \rightarrow P$ ,  $f : F \rightarrow P$ , and  $g : G \rightarrow P$ , and morphisms  $h : e \rightarrow f$  and  $k : f \rightarrow g$ , the composition of  $h$  and  $k$  is obtained by pasting the corresponding commutative triangles:

$$\begin{array}{ccccc} E & \xrightarrow{h} & F & \xrightarrow{k} & G \\ & \searrow e & \downarrow f & \swarrow g & \\ & & P & & \end{array}$$

From the commutativity of the individual triangles, one may infer that the ordinary composition of the maps  $h : E \rightarrow F$  and  $k : F \rightarrow G$  (the top edge) makes the outer triangle commute, meaning that  $h \circ k : e \rightarrow g$ . Finally, one should note that while  $\mathbf{Pos}$  is cartesian closed,  $\mathbf{Pos}/P$  need not be, although it does include a terminal object (the map  $1_P : P \rightarrow P$ ) as well as all binary products (defined via pullbacks, which we will consider presently). Still,  $\mathbf{Pos}/P$  contains a cartesian closed subcategory consisting of the fibrations over  $P$ ,  $\mathbf{Fib}(P)$ , on which we now focus our attention.

By way of introduction, we note that the fibrations to be described here (a special case of a vastly more general categorical construction) are entirely analogous to the more familiar ones that occur in the context of topology. As the reader may recall, a topological fibration over a space  $X$  is a space  $E$  together with a continuous map  $p : E \rightarrow X$  with the property that given any path  $\gamma$  between points  $x$  and  $x'$  in  $X$ , then for any choice of  $y' \in p^{-1}(x')$  there is a unique  $y \in p^{-1}(x)$  and path  $\gamma'$  from  $y$  to  $y'$  such that  $p(\gamma') = \gamma$ . In other words, every path in the base space lifts to a unique path in the fibred space, up to the choice of an endpoint. In the case of poset fibrations, the lifting property is essentially the same. The difference, of course, is that we are concerned not with the lifting of paths, but rather with the lifting of the only meaningful structure on a poset, namely the inequalities. Thus we make the following definition:

**Definition 4.1.1** *A fibration over a poset  $P$  is a monotone function  $e : E \rightarrow P$  that satisfies the following unique lifting property:*

*(ULP) Let  $p \leq p'$  in  $P$ . If  $q' \in e^{-1}(p')$ , there is a unique  $q \in e^{-1}(p)$  such that  $q \leq q'$ .*

In the ensuing discussion, we will often refer to the poset  $E$  itself as a fibration over  $P$ , provided that the mapping  $e : E \rightarrow P$  is clear from the context.

As we wish to consider the category  $\mathbf{Fib}(P)$  of fibrations over  $P$ , we require a suitable notion of the mappings between fibrations. We define them in the natural way: the morphisms between fibrations  $e : E \rightarrow P$  and  $f : F \rightarrow P$  are given by monotone functions  $h : E \rightarrow F$  such that  $e = h \circ f$ . In other words, a morphism between fibrations  $e : E \rightarrow P$  and  $f : F \rightarrow P$  is simply a morphism between  $e$  and  $f$  considered as objects of the slice category  $\mathbf{Pos}/P$ . Moreover, the identity maps and the operation of composition are precisely the same as those in  $\mathbf{Pos}/P$ . Hence, we note in passing,  $\mathbf{Fib}(P)$  is a full subcategory of  $\mathbf{Pos}/P$ .

If we were to proceed according to our usual method, we would now undertake a description of the cartesian closed structure on  $\mathbf{Fib}(P)$ . In this case, though, it is not entirely clear how this might be accomplished. Whereas, for example, the terminal object in  $\mathbf{Fib}(P)$  will likely be the identity fibration  $1_P : P \rightarrow P$ , and whereas it seems that the products in  $\mathbf{Fib}(P)$  will be identical to those in  $\mathbf{Pos}/P$ , it is by no means clear what the exponential of two fibrations might look like. As a result, it is best to postpone the description of the structures in  $\mathbf{Fib}(P)$  until we have proven a useful technical result, contained in Theorem 4.1.1 below, which asserts that the category  $\mathbf{Fib}(P)$  is equivalent to the presheaf category  $\mathbf{Set}^{P^{\text{op}}}$ , in the sense that there exists a pair of mutually (pseudo-)inverse functors  $\Phi_P : \mathbf{Set}^{P^{\text{op}}} \rightarrow \mathbf{Fib}(P)$  and  $\Psi_P : \mathbf{Fib}(P) \rightarrow \mathbf{Set}^{P^{\text{op}}}$ . As we will soon see, this equivalence will provide us with a particularly nice way of characterizing the structures in  $\mathbf{Fib}(P)$  in terms of the ones in  $\mathbf{Set}^{P^{\text{op}}}$ , with which we are by now quite familiar. More importantly, perhaps, it will follow as a trivial corollary to Theorem 4.1.1 that the functor  $\Phi_P : \mathbf{Set}^{P^{\text{op}}} \rightarrow \mathbf{Fib}(P)$ , which will be defined momentarily, is full, faithful, cartesian closed, and injective on objects. Hence in Section 4.2, when we specialize to the case in which  $P = \mathcal{O}(X_T)$ , the functor  $\Phi_{\mathcal{O}(X_T)} : \mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}} \rightarrow \mathbf{Fib}(\mathcal{O}(X_T))$  will furnish us with a proof that every  $\lambda$ -theory  $T$  has a representation in  $\mathbf{Fib}(\mathcal{O}(X_T))$ , namely the image under  $\Phi_{\mathcal{O}(X_T)}$  of the representation in  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ . But let us not get ahead of ourselves—we must first establish the aforementioned equivalence of categories:

**Theorem 4.1.1** *For any poset  $P$ , there is an equivalence of categories*

$$\mathbf{Set}^{P^{\text{op}}} \cong \mathbf{Fib}(P).$$

*In other words, there are functors*

$$\Phi_P : \mathbf{Set}^{P^{\text{op}}} \rightarrow \mathbf{Fib}(P) \text{ and } \Psi_P : \mathbf{Fib}(P) \rightarrow \mathbf{Set}^{P^{\text{op}}}$$

*such that there exist natural isomorphisms  $\Phi_P \circ \Psi_P \cong 1_{\mathbf{Fib}(P)}$  and  $\Psi_P \circ \Phi_P \cong 1_{\mathbf{Set}^{P^{\text{op}}}}$ .*

**Proof:** We begin by describing the two functors, starting with  $\Phi_P : \mathbf{Set}^{P^{\text{op}}} \rightarrow \mathbf{Fib}(P)$ . Given a presheaf  $F : P^{\text{op}} \rightarrow \mathbf{Set}$ , we define the image of  $F$  under  $\Phi_P$  to be the so-called category of elements of  $F$ , which we denote by  $\sum_{p \in P} F(p)$ . As the notation suggests, the category of elements should be regarded as a kind of disjoint union, albeit with a certain amount of additional structure encoded by its morphisms. Indeed, we define the objects of  $\sum_{p \in P} F(p)$  to be the elements of the

disjoint union of the sets  $F(p)$ , which we take to be  $\{(p, q) \mid p \in P, q \in F(p)\}$ . When there is no risk of confusion, though, we will dispense with the indexing, writing  $q \in F(p)$  in place of  $(p, q)$ . The morphisms of  $\sum_{p \in P} F(p)$  are defined as follows: given  $q \in F(p)$  and  $q' \in F(p')$ , we say that there is a morphism from  $q \rightarrow q'$  if  $p \leq p'$  and the map  $F(p \leq p') : F(p') \rightarrow F(p)$  satisfies  $F(p \leq p')(q') = q$ .

Of course, we must now check that the image under  $\Phi_P$  of any presheaf on  $P$  is in fact a fibration over  $P$ . It should be clear from the description of the arrows of  $\sum_{p \in P} F(p)$  that given any  $q \in F(p)$  and  $q' \in F(p')$ , there is an arrow from  $q$  to  $q'$  only if  $p \leq p'$ , in which case the morphism is unique. This means, of course, that  $\text{Hom}(q, q')$  is of cardinality zero or one. Thus we may define an order relation  $\leq$  on the objects of  $\sum_{p \in P} F(p)$  in the usual way, according to the condition that  $q \leq q'$  if and only if there exists an arrow from  $q$  to  $q'$ . One can easily check that this relation is reflexive, transitive, and antisymmetric, and therefore defines a partial ordering on  $\sum_{p \in P} F(p)$ . For example, since  $P$  is a poset, every  $p \in P$  satisfies the inequality  $p \leq p$ , which corresponds to the identity morphism  $1_p : p \rightarrow p$ . Thus, since  $F$  is a functor, it must be the case that  $F(p \leq p)$  is the identity arrow on  $F(p)$ . Naturally, this means that each  $q \in F(p)$  is mapped to itself under  $F(p \leq p)$ , which in turn implies that every  $q$  satisfies  $q \leq q$ . The transitivity and antisymmetry of the relation may also be derived from the order relation on  $P$  and the functorial nature of  $F$ . At any rate, it can be shown that  $\sum_{p \in P} F(p)$  is a poset. We must now verify that  $\sum_{p \in P} F(p)$  is an object of the slice category  $\mathbf{Pos}/P$  and, moreover, that it satisfies the universal lifting property described above.

To begin, we note that there is an obvious projection map  $\pi_F : \sum_{p \in P} F(p) \rightarrow P$  which sends each  $q \in F(p)$  to  $p \in P$ . It should be clear that  $\pi_F$  is monotone since, by definition of the relation on  $\sum_{p \in P} F(p)$ ,  $q \in F(p)$  is less than  $q' \in F(p')$  only if  $p \leq p'$  or, equivalently,  $\pi_F(q) \leq \pi_F(q')$ . At the very least, then, it is certainly true that  $\pi_F : \sum_{p \in P} F(p) \rightarrow P$  is an object of the slice category  $\mathbf{Pos}/P$ . Furthermore, one can show that  $\pi_F$  is a fibration. To that end, let  $p$  and  $p'$  be elements of  $P$ ,  $p \leq p'$ , and let  $q'$  be an element of  $F(p')$ . As we know,  $F$  carries the  $P$ -arrow  $p \leq p'$  to a set mapping  $F(p \leq p') : F(p') \rightarrow F(p)$ . It should be clear from the definition of the order relation on  $\sum_{p \in P} F(p)$  that  $q = F(p \leq p')(q')$  is the unique element of  $F(p)$  that satisfies the inequality  $q \leq q'$ . Naturally, this means that  $\pi_F : \sum_{p \in P} F(p) \rightarrow P$  satisfies the ULP, and is thus a fibration.

To show that the map  $\Phi_P$  described thus far extends to a functor from  $\mathbf{Set}^{P^{\text{op}}}$  to  $\mathbf{Fib}(P)$ , we must specify its action on the arrows of  $\mathbf{Set}^{P^{\text{op}}}$ , i.e. on natural

transformations between presheaves. We do so in the following manner: if  $\mu : F \rightarrow G$  is a natural transformation, then we define  $\Phi_P(\mu)$  to be the map between  $\sum_{p \in P} F(p)$  and  $\sum_{p \in P} G(p)$  which assigns to each  $q \in F(p)$  the element  $\mu_p(q) \in G(p)$ . Naturally, we must now check that  $\Phi_P(\mu)$  is in fact a **Fib**( $P$ )-arrow between the fibrations  $\pi_F : \sum_{p \in P} F(p) \rightarrow P$  and  $\pi_G : \sum_{p \in P} G(p) \rightarrow P$ , i.e. that  $\Phi_P(\mu)$  is a monotone function such that  $\pi_F = \pi_G \circ \Phi_P(\mu)$ .

The monotonicity of  $\Phi_P(\mu)$  follows easily from the naturality of  $\mu$ . In particular, let  $q \in F(p)$  and  $q' \in F(p')$ , and suppose that they satisfy the inequality  $q \leq q'$  when regarded as elements of  $\sum_{p \in P} F(p)$ . By definition of the order relation on  $\sum_{p \in P} F(p)$ , it follows that  $p \leq p'$  and that  $F(p \leq p') : F(p') \rightarrow F(p)$  maps  $q'$  to  $q$ . Now, by naturality of  $\mu$ , the following diagram must commute:

$$\begin{array}{ccc} F(p') & \xrightarrow{\mu_{p'}} & G(p') \\ \downarrow F(p \leq p') & & \downarrow G(p \leq p') \\ F(p) & \xrightarrow{\mu_p} & G(p) \end{array}$$

By what we have just said, if we traverse the diagram via the left and bottom edges,  $q'$  is mapped to  $q$ , and then to  $\mu_p(q)$ . Along the upper and right edges,  $q'$  is first mapped to  $\mu_{p'}(q')$ , then to some element  $G(p \leq p')(\mu_{p'}(q'))$  of  $G(p)$ . By definition of the order relation on  $\sum_{p \in P} G(p)$ , this element is less than or equal to  $\mu_{p'}(q')$ . Since the diagram is commutative, it must be the case that  $\mu_p(q) = G(p \leq p')(\mu_{p'}(q'))$ , and thus  $\mu_p(q) \leq \mu_{p'}(q')$ . This means, of course, that  $\Phi_P(\mu)(q) \leq \Phi_P(\mu)(q')$ . Hence  $\Phi_P(\mu)$  is indeed monotone.

Now, to complete the proof that  $\Phi_P(\mu)$  is a **Fib**( $P$ )-arrow from  $\pi_F$  to  $\pi_G$ , it remains only to show that  $\Phi_P(\mu)$  satisfies  $\pi_F = \pi_G \circ \Phi_P(\mu)$ . For any  $q \in F(p)$ , it is obviously the case that  $\Phi_P(\mu)(q)$  is an element of  $G(p)$ , which implies that  $\pi_G(\Phi_P(\mu)(q)) = p$ . Naturally,  $\pi_F(q) = p$  as well. Thus  $\Phi_P(\mu)$  is indeed a morphism of fibrations and, as a result, we have that  $\Phi_P$  is functorial from  $\mathbf{Set}^{P^{\text{op}}}$  to **Fib**( $P$ ).

We now construct the functor  $\Psi_P : \mathbf{Fib}(P) \rightarrow \mathbf{Set}^{P^{\text{op}}}$ , which is to be the inverse of  $\Phi_P$  (up to isomorphism). For any fibration  $f : F \rightarrow P$ , we define  $\Psi_P(f)$  to be the functor that assigns to each  $p \in P$  the fibre of  $f$  over  $p$ , i.e.  $\Psi_P(f)(p) = f^{-1}(p)$ . In addition, the image under  $\Psi_P(f)$  of any inequality  $p \leq p'$  in  $P$  is defined to be the set mapping  $\Psi_P(f)(p \leq p') : f^{-1}(p') \rightarrow f^{-1}(p)$  which carries each  $q' \in f^{-1}(p')$  to the unique  $q \in f^{-1}(p)$  that is less than  $q'$ , the existence of which is guaranteed

by the universal lifting property in  $\mathbf{Fib}(P)$ . Clearly,  $\Psi_P(f)$  is a presheaf on  $P$ . We must also account for the action of  $\Psi_P$  on the morphisms of  $\mathbf{Fib}(P)$ . Given a map  $h$  between fibrations  $e : E \rightarrow P$  and  $f : F \rightarrow P$ ,  $\Psi_P(h)$  is defined to be a natural transformation from  $\Psi_P(e)$  to  $\Psi_P(f)$ , the components of which are the set mappings  $(\Psi_P(h))_p : \Psi_P(e)(p) \rightarrow \Psi_P(f)(p)$  that assign to each  $x \in \Psi_P(e)(p) = e^{-1}(p)$  the value  $h(x) \in \Psi_P(f)(p) = f^{-1}(p)$ . To see that  $h(x) \in f^{-1}(p)$ , simply notice that  $h$  must satisfy  $e = f \circ h$ . As a result, if  $x \in e^{-1}(p)$ , then  $e(x) = p$ , which implies that  $f(h(x)) = p$  or, equivalently, that  $h(x) \in f^{-1}(p)$ .

It is not terribly difficult to prove that  $\Phi_P$  and  $\Psi_P$  are mutually inverse. First, let us show that  $\Psi_P(\Phi_P(F)) \cong F$  for any presheaf  $F$ : if  $p$  is an element of  $P$ , then

$$\begin{aligned} (\Psi_P(\Phi_P(F)))(p) &= (\Psi_P(\pi_F))(p) \\ &= \pi_F^{-1}(p) \\ &\cong F(p) \end{aligned}$$

since, by definition of  $\pi_F$ , the collection of elements of  $\sum_{p \in P} F(p)$  that project onto  $p$  is  $\pi_F^{-1}(p) = \{(p, q) \mid q \in F(p)\}$ , which is clearly isomorphic to  $F(p)$ . In particular,  $(\Psi_P(\Phi_P(F)))(p)$  is isomorphic to  $F(p)$  via the map  $\alpha_{F,p} : (\Psi_P(\Phi_P(F)))(p) \rightarrow F(p)$  defined by the rule  $\alpha_{F,p}(p, q) = q$  for all  $(p, q) \in \pi_F^{-1}(p)$ . To show that  $\Psi_P(\Phi_P(F))$  and  $F$  are isomorphic as presheaves, though, we must show that the mappings  $\alpha_{F,p}$  are the components of a natural transformation  $\alpha_F : \Psi_P(\Phi_P(F)) \rightarrow F$ . In other words, our task is to prove that for any  $P$ -arrow  $p \leq p'$ , the following diagram commutes:

$$\begin{array}{ccc} \Psi_P(\Phi_P(F))(p') & \xrightarrow{\alpha_{F,p'}} & F(p') \\ \downarrow (\Psi_P(\Phi_P(F)))(p \leq p') & & \downarrow F(p \leq p') \\ \Psi_P(\Phi_P(F))(p) & \xrightarrow{\alpha_{F,p}} & F(p) \end{array}$$

To that end, let  $(p', q')$  be an element of  $(\Psi_P(\Phi_P(F)))(p')$ . Its image under  $\alpha_{F,p'}$  is clearly the element  $q'$  of  $F(p')$ , which in turn is mapped to  $F(p \leq p')(q')$  in  $F(p)$ . To compute the image of  $(p', q')$  along the left and bottom paths, we first recall that

$$(\Psi_P \Phi_P F)(p \leq p') = (\Psi_P(\pi_F))(p \leq p'),$$

where the right hand side of the equation above is the map from  $\pi_F^{-1}(p')$  to  $\pi_F^{-1}(p)$ , that (setting aside the indexing for the moment) assigns to each  $q'$  in  $F(p')$  the unique

$q$  in  $F(p)$  such that  $q \leq q'$ . By the definition of the order relation on  $\sum_{p \in P} F(p)$ , though, the only  $q$  with this property is precisely the image of  $q'$  under  $F(p \leq p')$ . Expressed with the indices appropriate to  $q$  and  $q'$  as elements of the disjoint union  $\sum_{p \in P} \pi_F(p)$ , this simply means that  $((\Psi_P \Phi_P F)(p \leq p'))(p', q') = (p, F(p \leq p')(q'))$ . Of course, the image of this element under  $\alpha_{F,p'}$ , the bottom edge of the diagram, is  $F(p \leq p')$ . Hence the diagram commutes, and  $\Psi_P(\Phi_P(F))$  and  $F$  are naturally isomorphic via  $\alpha_F$ .

Now, to demonstrate that the functor  $\Psi_P \circ \Phi_P$  is naturally isomorphic to the identity functor on  $\mathbf{Set}^{P^{\text{op}}}$ ,  $1_{\mathbf{Set}^{P^{\text{op}}}}$ , we need only show that the isomorphisms  $\alpha_F : \Psi_P(\Phi_P(F)) \rightarrow F$  are the components of a natural transformation  $\alpha : \Psi_P \circ \Phi_P \rightarrow 1_{\mathbf{Set}^{P^{\text{op}}}}$ . In other words, we must show that for any  $F$  and  $G$  in  $\mathbf{Set}^{P^{\text{op}}}$  and any  $\mu : P \rightarrow Q$ , the following diagram commutes:

$$\begin{array}{ccc} \Psi_P(\Phi_T(F)) & \xrightarrow{\alpha_F} & F \\ \Psi_P(\Phi_T(\mu)) \downarrow & & \downarrow \mu \\ \Psi_P(\Phi_P(G)) & \xrightarrow{\alpha_G} & G \end{array}$$

As one can readily see, the commutativity of the square above is equivalent to that of the following:

$$\begin{array}{ccc} \Psi_P(\Phi_T(F))(p) & \xrightarrow{\alpha_{F,p}} & F(p) \\ (\Psi_P(\Phi_T(\mu)))_p \downarrow & & \downarrow \mu_p \\ \Psi_P(\Phi_P(G))(p) & \xrightarrow{\alpha_{G,p}} & G(p) \end{array}$$

Let us begin by considering the nature of the map  $\Psi_P(\Phi_T(\mu)) : \Psi_P(\Phi_T(F)) \rightarrow \Psi_P(\Phi_T(G))$ . By definition,  $\Phi_P(\mu)$  is the morphism between  $\pi_F : \sum_{p \in P} F(p) \rightarrow P$  and  $\pi_G : \sum_{p \in P} G(p) \rightarrow P$  that (suppressing indices) takes each  $q$  in the fiber of  $\pi_F$  over  $p \in P$  to the element  $\mu_p(q)$  of the fiber of  $\pi_G$  over  $p$ . The image of this map under  $\Psi_P$ , then, will be the natural transformation from  $\Psi_P(\pi_F)$  to  $\Psi_P(\pi_G)$  whose components  $(\Psi_P(\Phi_T(\mu)))_p$  are the maps from  $\pi_F^{-1}(p)$  to  $\pi_G^{-1}(p)$  that assigns to each  $q \in F(p)$  the element  $\Phi_P(\mu)(q) = \mu_p(q)$  of  $G(p)$ . Since all of this takes place in the disjoint unions  $\sum_{p \in P} \pi_F^{-1}(p)$  and  $\sum_{p \in P} \pi_G^{-1}(p)$ , though, we would do well to express it in the proper form, complete with indices:  $(\Psi_P(\Phi_T(\mu)))_p$  sends each  $(p, q)$

to  $(p, \mu_p(q))$ . After all that grueling work, it should now be clear that the diagram above commutes, since for any  $(p, q)$  in  $\Psi_P(\Phi_T(F))(p)$ ,

$$(\alpha_{G,p} \circ (\Psi_P(\Phi_T(\mu)))_p)(p, q) = \alpha_{G,p}(p, \mu_p(q)) = \mu_p(q) = (\mu_p \circ \alpha_{G,p})(p, q).$$

Thus  $\Psi_P \circ \Phi_P \cong 1_{\mathbf{Set}^{P^{\text{op}}}}$ .

To complete the proof that  $\Phi_P$  and  $\Psi_P$  are mutually inverse, we must show that an analogous result holds for the composition  $\Phi_P \circ \Psi_P$ , namely that it is isomorphic to the identity functor on  $\mathbf{Fib}(P)$ ,  $1_{\mathbf{Fib}(P)}$ . As a start, we prove that for any fibration  $e : E \rightarrow P$ , the fibration  $\Phi_P(\Psi_P(e)) : \sum_{p \in P} \Psi_P(e)(p) \rightarrow P$  is isomorphic to  $e$ . In other words, we seek to establish the existence of an isomorphism between  $\sum_{p \in P} \Psi_P(e)(p)$  and  $E$  that commutes with  $\Phi_P(\Psi_P(e))$  and  $e$  in the manner described above. Notice, first, that if we temporarily disregard the order relations on the posets  $E$  and  $\sum_{p \in P} \Psi_P(e)(p)$ , regarding them merely as sets, then

$$\begin{aligned} \Phi_P(\Psi_P(e)) &= \sum_{p \in P} \Psi_P(e)(p) \\ &= \sum_{p \in P} e^{-1}(p) \\ &\cong E \end{aligned}$$

It seems, then, that the underlying sets of  $E$  and  $\sum_{p \in P} \Psi_P(e)(p)$  are isomorphic, this time via the map  $\beta_e : \sum_{p \in P} \Psi_P(e)(p) \rightarrow E$  defined by  $\beta_e(p, q) = q$ . This is not exactly what we are looking for, of course, as we must demonstrate that  $E$  and  $\sum_{p \in P} \Psi_P(e)(p)$  are isomorphic not as sets, but rather as posets. To establish this point, we need only verify that  $\beta_e$  and  $\beta_e^{-1}$  are monotone or, equivalently, that for any  $q$  and  $q'$ ,  $q \leq q'$  in  $E$  if and only if (suppressing indices)  $q \leq q'$  in  $\sum_{p \in P} \Psi_P(e)(p)$  as well. To that end, suppose that  $q \leq q'$  in  $E$ , where  $q \in e^{-1}(p)$  and  $q' \in e^{-1}(p')$ . Since  $e$  is monotone, it must be the case that  $p \leq p'$ . Given this inequality, together with the fact that  $q'$  is an element of  $\Psi_P(e)(p')$ , the universal lifting property of  $\Phi_P(\Psi_P(e)) : \sum_{p \in P} \Psi_P(e)(p) \rightarrow P$  implies that there is a unique  $q''$  in  $\Psi_P(e)(p)$  such that  $q'' \leq q'$  in  $\sum_{p \in P} \Psi_P(e)(p)$ . In particular, our definition of  $\sum_{p \in P} \Psi_P(e)(p)$  forces  $q''$  to be  $\Psi_P(e)(p \leq p')(q')$ , which, as the reader may recall, is the unique element of  $e^{-1}(p)$  that is less than  $q'$  in  $E$ , namely  $q$ . From this observation, it follows that  $q \leq q'$  in  $\sum_{p \in P} \Psi_P(e)(p)$ . Moreover, it allows us to infer that if  $q \leq q'$  in  $\sum_{p \in P} \Psi_P(e)(p)$ , then  $q \leq q'$  in  $E$ . Hence  $\beta_e : \sum_{p \in P} \Psi_P(e)(p) \rightarrow E$  is an isomorphism of posets. Moreover, since it obviously commutes with the maps  $\Phi_P(\Psi_P(e))$  and  $e$ , it is also an isomorphism between the corresponding fibrations.

Finally, we need to show that the isomorphisms  $\beta_e$  are the components of a natural transformation  $\beta : \Phi_P \circ \Psi_P \rightarrow \mathbf{1}_{\mathbf{Fib}(P)}$ , from which it will follow that  $\Phi_P \circ \Psi_P \cong \mathbf{1}_{\mathbf{Fib}(P)}$ . This amounts to checking that the following diagram commutes for any fibration morphism  $h : e \rightarrow f$ :

$$\begin{array}{ccc} \Phi_P(\Psi_P)(e) & \xrightarrow{\beta_e} & e \\ \Phi_P(\Psi_P)(h) \downarrow & & \downarrow h \\ \Phi_P(\Psi_P)(f) & \xrightarrow{\beta_f} & f \end{array}$$

As before, we begin by considering the most complicated morphism:  $\Phi_P(\Psi_P(h))$ . By our earlier definitions, it is the map from  $\sum_{p \in P} \Psi_P(e)(p) = \sum_{p \in P} e^{-1}(p)$  to  $\sum_{p \in P} \Psi_P(f)(p) = \sum_{p \in P} f^{-1}(p)$  that assigns to each  $q \in e^{-1}(p)$  the element  $\Psi_P(h)(q) = h(q)$  of  $f^{-1}(p)$ . More formally,  $\Phi_P(\Psi_P(h))$  maps  $(p, q)$  in  $\sum_{p \in P} e^{-1}(p)$  to  $(p, h(q))$  in  $\sum_{p \in P} f^{-1}(p)$ . Clearly, then,

$$(\beta_f \circ \Phi_P(\Psi_P(h)))(p, q) = \beta_f(p, h(q)) = h(q) = (h \circ \beta_e)(p, q),$$

which means that the diagram does in fact commute.  $\square$

Before we return to our central preoccupation, the fibration representation, we must explore the cartesian closed structure of  $\mathbf{Fib}(P)$ . The terminal object, at least, is quite straightforward: for any poset  $P$ , the terminal object of  $\mathbf{Fib}(P)$  is the identity fibration  $1_P : P \rightarrow P$ , since for any  $e : E \rightarrow P$  in  $\mathbf{Fib}(P)$ , the only possible morphism  $!_e : e \rightarrow 1_P$  is the map  $!_e : E \rightarrow P$  such that  $e = 1_P \circ !_e$ , namely  $!_e = e$ . Hence every fibration  $e : E \rightarrow P$  has a unique map  $!_e : e \rightarrow 1_P$ . As noted above, however, when we consider  $\mathbf{Fib}(P)$  in isolation, it is less clear what the products and exponents should be. Using the equivalence above, however, we are able to translate our earlier definitions of presheaf products and exponentials into descriptions of the corresponding structures in  $\mathbf{Fib}(P)$ . Along the way, of course, we use the fact that  $\Psi_P$  and  $\Phi_P$  preserve products, exponentials, and any other structures defined by universal property, which follows easily from their role in the equivalence of categories. Thus, if  $e : E \rightarrow P$  and  $f : F \rightarrow P$  are fibrations, the

product  $e \times f$  is the image under  $\Phi_P$  of the presheaf product  $\Psi_P(e) \times \Psi_P(f)$ , namely

$$\begin{aligned}
 e \times f &= \Phi_P(\Psi_P(e) \times \Psi_P(f)) \\
 &= \sum_{p \in P} (\Psi_P(e) \times \Psi_P(f))(p) \\
 &= \sum_{p \in P} (\Psi_P(e)(p) \times \Psi_P(f)(p)) \\
 &= \sum_{p \in P} (e^{-1}(p) \times f^{-1}(p)),
 \end{aligned}$$

equipped with the obvious projection from  $\sum_{p \in P} (e^{-1}(p) \times f^{-1}(p))$  down to  $P$ . Experienced readers may recognize the poset  $\sum_{p \in P} (e^{-1}(p) \times f^{-1}(p))$  as the pullback of the corner of arrows  $E \xrightarrow{e} P \xleftarrow{f} F$ , the existence of which is guaranteed by the fact that **Pos** is cartesian closed. Diagrammatically,  $F \times_P E$  is the subset of  $E \times F$  with the property that the following diagram commutes:

$$\begin{array}{ccc}
 E \times_P F & \xrightarrow{p_2} & F \\
 p_1 \downarrow & & \downarrow f \\
 E & \xrightarrow{g} & P
 \end{array}$$

where  $p_1 : E \times_P F \rightarrow E$  and  $p_2 : E \times_P F \rightarrow F$  are the canonical projection mappings. In concrete terms,  $E \times_P F = \{(x, y) \in E \times F \mid e(x) = f(y)\}$ , subject to the order relation inherited from  $E \times F$ . This poset is a fibration over  $P$  via the map  $g \circ p_1 : E \times_P F \rightarrow P$  (or, equivalently, via  $f \circ p_2$ ). As one may already have guessed, given any fibrations  $e : E \rightarrow P$  and  $f : F \rightarrow P$ , the projections  $e \times f \rightarrow e$  and  $e \times f \rightarrow f$  are given by  $p_1 : E \times_P F \rightarrow E$  and  $p_2 : E \times_P F \rightarrow F$ , respectively. Moreover, given any fibration  $m : M \rightarrow P$  with arrows  $h : m \rightarrow e$  and  $k : m \rightarrow f$ , the pairing  $(h, k) : M \rightarrow E \times_P F$  is defined fibrewise. In particular, the restriction of  $(h, k)$  to a particular fiber  $m^{-1}(p)$ , denoted  $(h, k)_p$ , is defined to be the pairing  $(h_p, k_p) : m^{-1}(p) \rightarrow e^{-1}(p) \times f^{-1}(p)$ , where  $h_p$  and  $k_p$  denote the restrictions of  $h$  and  $k$  to  $m^{-1}(p)$ . One can easily check that the projections and the pairing operations interact in the correct manner.

Now that we have established the existence and nature of product objects in **Fib**( $P$ ), we must turn our attention to exponentials of fibrations. As above, given any fibrations  $f : F \rightarrow P$  and  $e : E \rightarrow P$ , we calculate the exponential  $f^e$  using the

equivalence of categories  $\Phi_P : \mathbf{Set}^{P^{\text{op}}} \rightleftarrows \mathbf{Fib}(P) : \Psi_P$  and our earlier definition of presheaf exponentials:

$$\begin{aligned} f^e &= \Phi_P(\Psi_P(f)^{\Psi_P(e)}) \\ &= \sum_{p \in P} \text{Hom}_{\mathbf{Set}^{P^{\text{op}}}}(y(p) \times \Psi_P(e), \Psi_P(f)) \end{aligned}$$

Since  $\Phi_P$  is full and faithful, it induces a bijection

$$\text{Hom}_{\mathbf{Set}^{P^{\text{op}}}}(y(p) \times \Psi_P(e), \Psi_P(f)) \cong \text{Hom}_{\mathbf{Fib}(P)}(\Phi_P(y(p) \times \Psi_P(e)), \Phi_P(\Psi_P(f)))$$

Thus

$$\begin{aligned} f^e &= \sum_{p \in P} \text{Hom}_{\mathbf{Fib}(P)}(\Phi_P(y(p) \times \Psi_P(e)), \Phi_P(\Psi_P(f))) \\ &= \sum_{p \in P} \text{Hom}_{\mathbf{Fib}(P)}(\Phi_P(y(p)) \times \Phi_P(\Psi_P(e)), \Phi_P(\Psi_P(f))) \end{aligned}$$

In order to reduce the right hand side of the equation above to a more manageable form, we begin by noticing that since  $\Phi_P$  and  $\Psi_P$  are inverse, it must be the case that  $\Phi_P(\Psi_P(e)) = e$  and  $\Phi_P(\Psi_P(f)) = f$ . We may also simplify the expression  $\Phi_P(y(p))$ . As the reader will recall from Chapter 2,  $y(p)$  is the contravariant Hom functor  $\text{Hom}_P(-, p)$  on  $P$ , which assigns to each  $p'$  in  $P$  the set  $\text{Hom}_P(p', p)$ . Of course, since  $P$  is a poset,  $\text{Hom}_P(p', p)$  is a one element set if  $p' \leq p$ , and is empty otherwise. In essence, then,  $\Phi_P(y(p)) = \sum_{p' \in P} \text{Hom}_P(p', p)$  is simply the subposet of  $P$  consisting of all  $p' \in P$  such that  $p' \leq p$ , henceforth denoted by  $P/p$  (as the notation suggests, this is another example of a slice category). The poset  $P/p$  is a fibration via the obvious inclusion mapping  $\iota : P/p \rightarrow P$ , as one can readily verify.

Plugging these results back into the the equation above, we find that

$$f^e = \sum_{p \in P} \text{Hom}_{\mathbf{Fib}(P)}(\iota \times e, f)$$

As we've seen, the product  $\iota \times e$  is the pullback of  $e : E \rightarrow P$  along  $\iota : P/p \rightarrow P$ , the map  $p^*e$  that makes the following diagram commute:

$$\begin{array}{ccc} p^*E & \longrightarrow & E \\ \downarrow p^*e & & \downarrow e \\ P/p & \xrightarrow{\quad \iota \quad} & P \end{array}$$

where  $p^*E = \sum_{p' \in P/p} e^{-1}(p')$ , the disjoint union of the fibers of  $e$  above  $p' \in P/p$ , and the map from  $p^*E$  to  $E$  is the obvious inclusion. To put it more simply,  $p^*e$  is the restriction of  $e$  to the fibers above  $p' \in P/p$ . It should be clear that  $p^*e : p^*E \rightarrow P/p$  is a fibration over  $P/p$ , i.e. an object of  $\mathbf{Fib}(P/p)$ . As such, the Hom-sets whose disjoint union constitutes  $f^e$  must be taken in  $\mathbf{Fib}(P/p)$  rather than  $\mathbf{Fib}(P)$ , and the codomain  $f$  must be replaced by the suitable fibration over  $P/p$ , namely  $p^*f : p^*F \rightarrow P/p$ . In short,

$$f^e = \sum_{p \in P} \text{Hom}_{\mathbf{Fib}(P/p)}(p^*e, p^*f),$$

with the obvious projection from  $\sum_{p \in P} \text{Hom}_{\mathbf{Fib}(P/p)}(p^*e, p^*f)$  to  $P$  that takes any  $h \in \text{Hom}_{\mathbf{Fib}(P/p)}(p^*e, p^*f)$  to  $p$ . For the sake of convenience, we henceforth refer to  $\text{Hom}_{\mathbf{Fib}(P/p)}(p^*e, p^*f)$  as  $(f^e)^{-1}(p)$ .

We must now specify the form of the evaluation map  $\epsilon : f^e \times e \rightarrow f$ . First, we notice that, on the basis of the foregoing discussion,  $f^e \times e$  is the obvious projection from  $\sum_{p \in P} ((f^e)^{-1}(p) \times e^{-1}(p))$  to  $P$ , say  $e \circ p_2$ . We define the evaluation map  $\epsilon$  as follows: given an arbitrary element  $(h, q)$  of  $\text{Hom}_{\mathbf{Fib}(P/p)}(p^*e, p^*f) \times e^{-1}(p)$ , the fibre of  $f^e \times e$  over  $p$ , let  $\epsilon(h, q) = h(q)$ , which is clearly an element of the fiber of  $f$  over  $p$ . The mapping  $\epsilon$  thus defined is monotone, as one can easily check. In particular, if  $(h, q) \leq (h', q')$  in  $\sum_{p \in P} ((f^e)^{-1}(p) \times e^{-1}(p))$ , then by virtue of the definition of the order relation on products,  $h \leq h'$  and  $q \leq q'$ . Hence, from the monotonicity of  $h$  and the definition of the ordering on such functions, we may infer that

$$\epsilon(h, q) = h(q) \leq h(q') \leq h'(q') = \epsilon(h', q').$$

It now remains only to show that  $\epsilon : \sum_{p \in P} ((f^e)^{-1}(p) \times e^{-1}(p)) \rightarrow \sum_{p \in P} f^{-1}(p)$  gets along with the maps  $e \circ p_2 : \sum_{p \in P} ((f^e)^{-1}(p) \times e^{-1}(p)) \rightarrow P$  and  $f : \sum_{p \in P} f^{-1}(p) \rightarrow P$ . But this is rather trivial: since  $q \in e^{-1}(p)$  and  $h(q) \in f^{-1}(p)$ ,

$$(e \circ p_2)(h, q) = e(q) = p = f(h(q)) = (f \circ \epsilon)(h, q),$$

as required. Thus, it seems, this  $\epsilon : f^e \times e \rightarrow f$  is a suitable evaluation map.

Finally, given this definition of the evaluation arrow, there is no particular difficulty in characterizing the operation of transposition. To begin, let  $e : E \rightarrow P$ ,  $f : F \rightarrow P$ , and  $g : G \rightarrow P$  be fibrations, and let  $h : g \times e \rightarrow f$  be a morphism of fibrations. Our task is to construct a morphism  $\tilde{h} : g \rightarrow f^e$  that makes the following

diagram commute:

$$\begin{array}{ccc}
 f^e \times e & \xrightarrow{\epsilon} & f \\
 \tilde{h} \times 1_e \uparrow & & \nearrow h \\
 g \times e & & 
 \end{array}$$

Of course, our choice of the transpose  $\tilde{h}$  is essentially forced by this commutativity condition:  $\tilde{h}$  must be the map from  $\sum_{p \in P} g^{-1}(p)$  to  $\sum_{p \in P} (f^e)^{-1}(p)$  that assigns to each  $q \in g^{-1}(p)$  the function  $\tilde{h}(q) \in (f^e)^{-1}(p)$  defined by the rule that  $\tilde{h}(q)(q') = h(q, q')$  for all  $q' \in e^{-1}(p)$ . The proof that  $\tilde{h}$  possesses all of the properties required of a morphism between  $g$  and  $f^e$  is left as an exercise.

At any rate, we have now established that the category  $\mathbf{Fib}(P)$  is cartesian closed, and have obtained a detailed description of the products, the exponents, and the operations of pairing and transposition. As a result, we may subject it to our characteristic method of analysis, wherein notions of semantic completeness are reduced to the properties of cartesian closed functors.

## 4.2 The Fibration Representation

We are now in a position to prove our central result, the fibration representation theorem:

**Theorem 4.2.1** *Every  $\lambda$ -theory  $T$  has a representation in the category of fibrations over the poset of open subsets of a topological space. In particular, for every  $T$  there is a topological space  $X_T$  and a model  $\llbracket - \rrbracket : T \rightarrow \mathbf{Fib}(\mathcal{O}(X_T))$  satisfying the following conditions:*

1. *For any closed terms  $M$  and  $N$  in  $\mathcal{L}(T)$ ,*

$$\llbracket M \rrbracket = \llbracket N \rrbracket \text{ if and only if } T \vdash M = N$$

2. *Every arrow of the form  $f : 1 \rightarrow \llbracket \sigma \rrbracket$  in  $\mathbf{Fib}(P)$  is the interpretation of some closed term  $\vdash M : \sigma$  in  $\mathcal{L}(T)$ .*

**Proof:** It is clear, no doubt, how we must proceed. By the presheaf representation theorem of Section 3.3, we know that every  $\lambda$ -theory  $T$  has a representation in the

category  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ , where  $X_T$  is a topological space constructed from the syntax of  $T$  (in a manner that will be described presently). Given this knowledge, the fibration representation will follow if we are able to exhibit a full, faithful, cartesian closed functor from  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$  to  $\mathbf{Fib}(\mathcal{O}(X_T))$ . The functor  $\Phi_P$  that we constructed in the proof of Theorem 4.1.1 suggests itself as an obvious candidate, although we now specialize to the case in which  $P = \mathcal{O}(X_T)$ . As it turns out, the functor  $\Phi_{\mathcal{O}(X_T)}$  (henceforth abbreviated as  $\Phi_T$ ) is entirely sufficient for our purposes:

**Lemma 4.2.1** *The functor  $\Phi_T : \mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}} \rightarrow \mathbf{Fib}(\mathcal{O}(X_T))$  is full, faithful, cartesian closed, and injective on objects.*

**Proof:** As we saw in the proof of Theorem 4.1.1, there are natural transformations  $\alpha : \Psi_T \circ \Phi_T \rightarrow \mathbf{1}_{\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}}$  and  $\beta : \Phi_T \circ \Psi_T \rightarrow \mathbf{1}_{\mathbf{Fib}(\mathcal{O}(X_T))}$  whose components are isomorphisms. As a result, for any natural transformation  $\mu$  between presheaves  $P$  and  $Q$ , the following diagram commutes:

$$\begin{array}{ccc} \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_P} & P \\ \Psi_T(\Phi_T(\mu)) \downarrow & & \downarrow \mu \\ \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_Q} & Q \end{array}$$

Now, let  $\eta$  be another natural transformation from  $P$  to  $Q$ , noticing that, by naturality, it must satisfy an analogous commutative diagram. Clearly, if it is the case that  $\Phi_T(\mu) = \Phi_T(\eta)$ , then  $\Psi_T(\Phi_T(\mu)) = \Psi_T(\Phi_T(\eta))$ , which in turn is true only if the following diagram commutes:

$$\begin{array}{ccccc} P & \xleftarrow{\alpha_P} & \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_P} & P \\ \eta \downarrow & & \downarrow \Psi_T(\Phi_T(\mu)) & & \downarrow \mu \\ Q & \xleftarrow{\alpha_Q} & \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_Q} & Q \end{array}$$

But, as one can easily see, this implies that  $\mu = \eta$ . Hence  $\Phi_T$  is faithful. Moreover,  $\Psi_T$  is faithful as well, by symmetry.

Using this fact, one may now show that  $\Phi_T$  is also faithful. Consider an arbitrary fibration morphism  $h : \Phi_T(P) \rightarrow \Phi_T(Q)$ , where  $P$  and  $Q$  are some presheaves in

$\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$ . Now, let  $\mu : P \rightarrow Q$  be the morphism defined by  $\mu = \alpha_G^{-1} \circ (\Psi_T(h)) \circ \alpha_G$ , i.e. such that

$$\begin{array}{ccc} \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_P} & P \\ \Psi_T(h) \downarrow & & \downarrow \mu \\ \Psi_T(\Phi_T(Q)) & \xrightarrow{\alpha_Q} & Q \end{array}$$

commutes. Of course, by naturality of  $\alpha$ , the following diagram also commutes:

$$\begin{array}{ccc} \Psi_T(\Phi_T(P)) & \xrightarrow{\alpha_P} & P \\ \Psi_T(\Phi_T(\mu)) \downarrow & & \downarrow \mu \\ \Psi_T(\Phi_T(Q)) & \xrightarrow{\alpha_Q} & Q \end{array}$$

Taken together, the diagrams imply that  $\Psi_T(\Phi_T(\mu)) = \Psi_T(h)$ . Of course, since  $\Psi_T$  is faithful, this means that  $\Phi_T(\mu) = h$ . As one can easily see, it follows that  $\Phi_T$  is full. We have already seen that  $\Phi_T$  must be cartesian closed, again as a result of the equivalence contained in Theorem 4.1.1. The proof of the injectivity of  $\Phi_T$  on objects, another easy exercise in definitions, is left to the reader.  $\square$

By the usual argument, then, the image under  $\Phi_T$  of the representation of  $T$  in  $\mathbf{Set}^{\mathcal{O}(X_T)^{\text{op}}}$  is a representation of  $T$  in  $\mathbf{Fib}(\mathcal{O}(X_T))$ .  $\square$

One should not, by any means, take the brevity and ease of this argument as a sign of the triviality of the completeness result that it establishes. Indeed, it would more rightly be viewed as a testament to the power of our earlier results and, in particular, to that of the sheaf representation and spatial covering theorems, the complexities of which we have but sketched here. Regardless, we have now proven that every  $\lambda$ -theory  $T$  has a representation in  $\mathbf{Fib}(\mathcal{O}(X_T))$  and, as a result, that the class of models in categories of the form  $\mathbf{Fib}(\mathcal{O}(X_T))$  constitutes strongly complete semantics for the  $\lambda$ -calculus. Our only remaining task is to give a description of the fibration representation of a given  $\lambda$ -theory.

First, though, we must undertake an examination of the topological space  $X_T$  introduced in Section 3.2, once again relying on [1] as our basic source. Before we begin, there are a few preliminaries that must be dealt with. First of all, for every type  $\sigma$  of the theory  $T$ , we choose a distinct variable  $x_\sigma$  and, for the sake of

convenience, we fix the type assignment  $x_\sigma : \sigma$ . In a slight modification of our earlier notation, for any term  $M$  in  $\mathcal{L}(T)$ , we write  $M[x_\sigma]$  if there are no free variables in  $M$  other than  $x_\sigma$  or, equivalently, if  $\lambda x_\sigma.M$  is closed. For each type  $\sigma$ , we denote by  $\mathcal{L}[x_\sigma]$  the set of all such  $M[x_\sigma]$ . The points of the space  $X_T$  will be closely related to enumerations of these sets, a notion that we now recall:

**Definition 4.2.1** *An enumeration of  $\mathcal{L}[x_\sigma]$  is a surjective partial function  $f : \mathbf{N} \rightarrow \mathcal{L}[x_\sigma]$ . In other words,  $f$  is a map defined on some subset  $D_f$  of the natural numbers, and which has the property that every term  $M[x_\sigma]$  in  $\mathcal{L}[x_\sigma]$  is in the image of  $D_f$  under  $f$ .*

Finally, the construction of  $X_T$  requires a relation of equivalence between enumerations. Given  $(\sigma, f)$  and  $(\tau, g)$ , we say that  $(\sigma, f)$  is equivalent to  $(\tau, g)$  and write  $(\sigma, f) \sim (\tau, g)$  if  $D_f = D_g$  and there exist terms  $M[x_\sigma]$  and  $N[x_\tau]$  of types  $\tau$  and  $\sigma$ , respectively, such that the following equations hold in  $T$ :

$$\begin{aligned} x_\sigma &\vdash N[M/x_\tau] = x_\sigma \\ x_\tau &\vdash M[N/x_\sigma] = x_\tau \\ x_\sigma &\vdash g(i)[M/x_\tau] = f(i) \\ x_\tau &\vdash f(i)[N/x_\sigma] = g(i) \end{aligned}$$

As the equivalence relation thus defined is rather complex, we would do well to pause momentarily and attempt to obtain a better understanding of how it might work in practice. This may be most readily accomplished, perhaps, by considering a special case. Suppose, then, that two enumerations  $(\sigma, f)$  and  $(\tau, g)$  are equivalent and, moreover, that they agree in their first coordinate, i.e.  $\sigma = \tau$ . Then the first two equations in the definition of the equivalence relation become

$$x_\sigma \vdash M[N/x_\sigma] = x_\sigma \text{ and } x_\sigma \vdash N[M/x_\sigma] = x_\sigma$$

for some  $M[x_\sigma]$  and  $N[x_\sigma]$ , both of which are of type  $\sigma$  relative to the context  $x_\sigma$ . Taken together, these equations clearly imply that  $M = x_\sigma = N$ . As a result, the third equation becomes

$$x_\sigma \vdash g(i)[x_\sigma/x_\sigma] = f(i),$$

and thus  $x_\sigma \vdash f(i) = g(i)$  for all  $i \geq 0$ . In other words, enumerations  $(\sigma, f)$  and  $(\sigma, g)$  are equivalent only if  $f$  and  $g$  agree with respect to the context  $x_\sigma$ .

We are now prepared to describe the topological space  $X_T$ . To begin, we define its underlying point set as consisting of all equivalence classes of enumerations  $(\sigma, f)$ . Of course, since we are concerned not with the points of  $X_T$  but rather with its open sets, we must now turn our attention to the topological structure on  $X_T$ . First, though, we require an additional notion pertaining to the syntax of the  $\lambda$ -calculus:

**Definition 4.2.2** *We say that a term  $M[x_\sigma]$  is a substitution instance of  $N[x_\tau]$  if there exists a term  $S[x_\sigma]$  of type  $\tau$  such that*

$$x_\sigma \vdash M[x_\sigma] = N[S/x_\tau].$$

*When  $M[x_\sigma]$  is a substitution instance of  $N[x_\tau]$ , we write  $M[x_\sigma] \prec N[x_\tau]$ .*

We may use this relation to define the collection of basic open sets of  $X_T$  from which the topology on  $X_T$  is to be generated. In particular, for any  $n \geq 1$ , any  $n$ -tuple  $(k_i) = (k_1, \dots, k_n)$  of natural numbers, and any  $n$ -tuple of terms  $(M_i) = (M_1[x_{\sigma_1}], \dots, M_n[x_{\sigma_n}])$ , we designate the set

$$B_{(k_i), (M_i)} = \{(\sigma, f) \mid f(k_i) \prec M_i[x_{\sigma_i}]\}$$

as being open. Naturally, in order for the  $B_{(k_i), (M_i)}$  thus defined to make sense as subsets of  $X_T$ , it must be the case that they are not merely sets of enumerations  $(\sigma, f)$ , but rather that they are sets of points of  $X_T$ . In other words, these sets must be defined on equivalence classes of enumerations. The verification that the  $B_{(k_i), (M_i)}$  do indeed have this property is relatively easy. To begin, we let  $(\sigma, f)$  and  $(\tau, g)$  be enumerations, and suppose that  $(\sigma, f) \sim (\tau, g)$ . Now, suppose that  $(\sigma, f)$  is contained in some  $B_{(k_i), (M_i)}$ . Then for every  $i \leq n$ ,  $f(k_i) \prec M_i[x_{\sigma_i}]$ . This means, of course, that for every  $i \leq n$  there is a term  $S[x_\sigma]$  of type  $\sigma_i$  such that

$$x_\sigma \vdash f(k_i) = M_i[S/x_{\sigma_i}]$$

From the third equation in the definition of the equivalence relation on enumerations given above, we know that  $(\sigma, f) \sim (\tau, g)$  only if there exists a term  $N[x_\tau]$  of type  $\sigma$  such that

$$x_\tau \vdash f(i)[N/x_\sigma] = g(i)$$

for all  $i \geq 0$ . But since  $x_\sigma \vdash f(k_i) = M_i[S/x_{\sigma_i}]$ , as we noted above, one can see that

$$x_\tau \vdash g(k_i) = (M_i[S/x_{\sigma_i}])[N/x_\sigma]$$

and, since  $x_\sigma$  appears free only in the occurrences of  $S[x_\sigma]$  in  $M_i[S/x_{\sigma_i}]$ ,

$$x_\tau \vdash g(k_i) = M_i[S[N/x_{\sigma_i}]/x_{\sigma_i}].$$

Hence  $g(k_i) \prec M_i[x_{\sigma_i}]$  for all  $i \leq n$ , which means that  $(\tau, g) \in B_{(k_i), (M_i)}$ . Similarly (or better yet, by symmetry) if  $(\tau, g) \in B_{(k_i), (M_i)}$ , then so is  $(\sigma, f)$ . We have shown, then, that the  $B_{(k_i), (M_i)}$  are defined on equivalence classes of enumerations, as desired.

It should be clear that the collection  $\mathcal{B}$  of sets of the form  $B_{(k_i), (M_i)}$  does not itself constitute a topology on  $X_T$  since, for example, the set  $X_T$  will not be contained in  $\mathcal{B}$ . Nevertheless,  $\mathcal{B}$  is the basis for a topology on  $X_T$ , in the sense that there is a topology  $\mathcal{T}$  on  $X_T$  generated from the elements of  $\mathcal{B}$ , namely that in which a set  $U \subseteq X_T$  is open if and only if it is the union or finite intersection of elements of  $\mathcal{B}$ , i.e. of sets  $B_{(k_i), (M_i)}$ . As usual, we establish that the collection of  $B_{(k_i), (M_i)}$  is a topological basis by verifying that it has the following properties:

1. If  $(\sigma, f) \in X_T$ , then there is some  $B_{(k_i), (M_i)}$  containing  $x$ .
2. Given any  $B_{(k_i), (M_i)}$  and  $B_{(l_i), (N_i)}$ , if  $x$  is contained in their intersection, then there is some  $B_{(j_i), (L_i)}$  containing  $(\sigma, f)$  such that

$$B_{(j_i), (L_i)} \subseteq B_{(k_i), (M_i)} \cap B_{(l_i), (N_i)}.$$

It should be more or less obvious that the first condition is satisfied as, given any  $(\sigma, f)$  in  $X_T$ , it is trivially true that  $f(1) \prec f(1)[x_\sigma]$ , and thus it must be the case that  $(\sigma, f) \in B_{(1), (f(1)[x_\sigma])}$ . The second condition is also rather trivial. One need only notice that the intersection of any  $B_{(k_i), (M_i)}$  and  $B_{(l_j), (N_j)}$  is the set

$$B_{(k_i), (M_i)} \cap B_{(l_j), (N_j)} = \{(\sigma, f) \mid f(k_i) \prec M_i[x_{\sigma_i}], \text{ and } f(l_j) \prec N_j[x_{\tau_j}], i \leq n, j \leq m\},$$

which is clearly of the correct form to be an element of  $\mathcal{B}$ . In particular,

$$B_{(k_i), (M_i)} \cap B_{(l_j), (N_j)} = B_{(k_i, l_j), (M_i, N_j)}$$

where  $(k_i, l_j)$  and  $(M_i, N_j)$  denote the concatenations of the relevant sequences. Hence  $\mathcal{B}$  obviously satisfies the second condition, and must therefore be a topological basis. Now, let  $X_T$  be endowed with the topology  $\mathcal{T}$  generated from  $\mathcal{B}$ , and let  $\mathcal{O}(X_T)$  be the poset of open subsets, ordered by inclusion.

One could now proceed to give an explicit description of the representation  $\llbracket - \rrbracket$  of  $T$  in  $\mathbf{Fib}(\mathcal{O}(X_T))$ , specifying the fibration  $p_{\llbracket B \rrbracket} : \llbracket B \rrbracket \rightarrow \mathcal{O}(X_T)$  assigned to each

basic type  $B$  of  $T$ , and the fibration morphism  $\llbracket b : \tau \rrbracket : 1_{\mathcal{O}(X_T)} \rightarrow \llbracket \tau \rrbracket$  assigned to each basic term  $b : \tau$ , then generating the full representation in the manner described in Section 2.2. Indeed, modulo a few technical issues, anyone who has read and understood this thesis should have little difficulty translating the sheaf representation described in [1] into a representation in  $\mathbf{Fib}(\mathcal{O}(X_T))$  using the series of functors that we have considered here. Ultimately, though, one will find that neither the process nor the result is particularly enlightening. Moreover, such a preoccupation with the construction of the representation is not entirely consistent with the spirit of our enquiry, as our overriding concern has simply been to establish the strong completeness of, and existence of representations in, the system of semantics consisting of models in categories of poset fibrations (or, hearkening back to Section 3.3, in categories of presheaves on posets), for which purpose the abstract proofs of existence that we have presented here are entirely sufficient. The knowledge of the completeness of semantics in such simple and intuitively appealing categories should be reward enough.



# Bibliography

- [1] Steven Awodey. Topological representation of the lambda calculus. *Mathematical Structures in Computer Science*, 10:81–96, 2000.
- [2] H. P. Barendregt. *The lambda calculus: its syntax and semantics*. North-Holland, New York, NY, revised edition, 1984.
- [3] H. P. Barendregt. Functional programming and lambda calculus. In Jan Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 321–363. MIT Press, Cambridge, MA, 1990.
- [4] Carsten Butz and Ieke Moerdijk. Topological representation of sheaf cohomology of sites. Technical Report 973, Universiteit Utrecht, 1996.
- [5] Robert Goldblatt. *Topoi: the categorical analysis of logic*. North-Holland, New York, NY, revised edition, 1984.
- [6] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.
- [7] J. Roger Hindley. *Basic simple type theory*. Cambridge University Press, New York, NY, 1997.
- [8] C. J. Isham and J. Butterfield. Some possible roles for topos theory in quantum theory and quantum gravity. Submitted to Foundations of Physics for an issue in honor of Marisa Dalla Chiara, September 1999.
- [9] Joachim Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Cambridge University Press, New York, NY, 1986.
- [10] Saunders Mac Lane. *Categories for the working mathematician*. Springer-Verlag, New York, NY, second edition, 1998.
- [11] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory*. Springer-Verlag, New York, NY, 1992.
- [12] Colin McLarty. *Elementary categories, elementary toposes*. Oxford University Press, New York, NY, 1992.

- [13] John C. Mitchell. Type systems for programming languages. In Jan Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 365–458. MIT Press, Cambridge, MA, 1990.
- [14] John C. Mitchell and Eugenio Moggi. Kripke-style models for typed lambda calculus. *Annals of Pure and Applied Logic*, 51:99–124, 1991.
- [15] Dana S. Scott. Relating theories of the lambda calculus. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 403–450. Academic Press, New York, NY, 1980.