UNIVERSITY OF CALGARY

A Categorical Extension of the Curry-Howard Isomorphism

by

Joseph McDonald

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ARTS

GRADUATE PROGRAM IN PHILOSOPHY

CALGARY, ALBERTA

SEPTEMBER, 2019

Abstract

Within this work, we investigate some aspects of the interaction that occurs between category theory and the typed λ-calculus. In particular, we examine the equivalence of categories that can be demonstrated between the category of Cartesian closed categories and the category of typed λ-theories. As will be seen, a convenient technique that can be employed so as to make this interaction explicit is the application of algebraic theories (often known as "Lawvere theories"), which allow us to characterize various model-theoretical aspects of typed λ-theories in the abstract algebraic setting of a Cartesian closed category. What follows from this interaction between Cartesian closed categories and typed λ-theories is that of a categorical extension of the Curry-Howard isomorphism, which attributes a similar correspondence between the typed λ-calculus and the positive fragment of the intuitionistic propositional calculus.

Acknowledgments

I would like to start off by thanking my thesis supervisor Dr. Richard Zach for all his support and guidance thought this research project, as well as for his excellent graduate course offerings in mathematical logic. I would also like to thank Dr. Nicole Wyatt and Dr. Robin Cockett for sitting on my thesis committee, along with Dr. Bryson Brown, my undergraduate honours thesis supervisor, who sparked my passion for logic early on in my undergraduate degree. Lastly, I would like to thank my mother Wendy McDonald for her continued support throughout my studies.

Table of Contents

Chapter 0

Introduction

Among the variety of approaches that have been employed as a means to come to grips with some particular aspects of the foundations of mathematics, we have on the one hand, proof theory, which is largely tied up in the study of the intuitionistic propositional calculus as well as the typed λ-calculus. On the other hand, we have category theory, whose applications to logic have been found to be in part, tied up in the study of Cartesian closed categories.

Intuitionistic logic originally fell out of Brouwer's constructivist approach to mathematics which he initially formulated in his doctoral thesis "Over de Grondslagen der Wiskunde"[On the Foundations of Mathematics] (1907). According to Brouwer's constructivist program, the "correctness" of a mathematical statement is contingent upon the presence of some sort of algorithmic procedure which gives a construction or proof of the statement in question. A common misconception of intuitionistic logic is that it is a logical calculus which underlies intuitionistic constructive mathematics; rather, it is fact intuitionistic constructive mathematics which underlies intuitionistic logic and is built upon the application of the foundational principles of constructive intuitionistic mathematics to formal languages. “Intuitionistic [constructive] mathematics consists in the act of effecting mental constructions of a certain kind. These are themselves not linguistic in nature, but when acts of construction and their results are described in a language, the descriptions may come to exhibit linguistic patterns. Intuitionistic logic is the mathematical study of these patterns, and in particular of those that characterize valid inferences.”[1]

The first axiomatization of intuitionistic logic was given by Arend Heyting in his “Die Formalen Regeln der Intuitionistischen Logik” [The Formal Rules of Intuitionistic Logic I-III] (1930), which was then most notably reformulated as both a natural deduction calculus and sequent calculus by Gerhard Gentzen in his doctoral thesis “Untersuchungen über das Logische Schlieben I-II” [Investigations into Logical Deduction I-II] (1935a/1935b), which he proved to be normalizable and cut-eliminable, respectively. Generally speaking, intuitionistic logic is a system of classical logic minus the law of the excluded middle and the law of double negation elimination, which can be formulated by the following sequents

$$\Gamma \vdash \phi \vee \neg\phi, \;\; \Gamma \vdash \neg\neg\phi \Rightarrow \phi$$

respectively, for any formula ϕ and finite collection of formulas Γ. The failure of these theorems within the intuitionistic calculus (which are characteristic of the classical calculus) stems from the underlying constructivist interpretation of the logical connectives, which diverges from that of the classical interpretation. In the setting of a classical propositional calculus,

[1] See van Atten, Mark, "The Development of Intuitionistic Logic", 2017.

we interpret the connectives as follows:

- $\phi \wedge \psi$ means "both ϕ and ψ are true"
- $\phi \vee \psi$ means "ϕ is true or ψ is true"
- $\phi \Rightarrow \psi$ means "ϕ is false or ψ is true"
- $\perp$ is always false, i,.e., a contradiction
- $\neg\phi$ means "ϕ is false"

However, in the context of the intuitionistic propositional calculus, assuming we have in mind a well defined notion of what it means to have a construction of an atomic formula, we interpret the connectives as follows:

- A construction of $\phi_1 \wedge \phi_2$ is a pair $\langle C_1, C_2 \rangle$ such that C_1 is a construction of ϕ_1 and C_2 is a construction of ϕ_2
- A construction of $\phi_1 \vee \phi_2$ is a pair $\langle s, C \rangle$ such that s is 1 and C is a construction of ϕ_1 or s is 2 and C is a construction of ϕ_2
- A construction of $\phi \Rightarrow \psi$ is a function in which a construction of ϕ is transformed into a construction of ψ
- There exists no construction of $\perp$
- $\neg\phi$ is defined by $\phi \Rightarrow \perp$ so is a function which transforms a construction of ϕ into a construction of $\perp$

This interpretation of the connectives within the intuitionistic propositional calculus is commonly refereed to as the "Brouwer-Heyting-Kolmogorov interpretation" or simply the "BHK Interpretation."

For the purposes of this work, we represent the validity of such constructions of formulas as being given by Gentzen-style tree structures that can be extended by n applications of axioms and rules of inference belonging to the calculus. Below are two very generalized examples of such a structure:

$$\begin{array}{c} \phi \\ \hline \vdots \\ \\ \vdots \\ \hline \psi \end{array} \qquad\qquad \begin{array}{c} [\phi]^k \\ \hline \vdots \\ \\ \vdots \\ \hline \psi \end{array}\, k$$

Note that the leftmost tree represents a proof of ψ from ϕ, in which ϕ is an axiom and the rightmost tree represents a proof of ψ from ϕ, in which ϕ is an assumption that has been discharged at line $n-1$.

Around the same time, Alonzo Church was developing a computational proof system known as the λ-calculus, which he originally formulated in "A Set of Postulates for the Foundations of Logic" (1932). Generally speaking, the λ-calculus is a mathematical model of computation for calculating with functions. The λ-calculus starts from the observation that there are two basic operations we can perform on functions. First, there is the function-theoretical operation of *application*: Given a function f and an argument a, we typically write $f\,a$ to denote the the application of f to a in which a is an element in the codomain of f. Second, the λ-calculus is grounded upon the function-theoretical operation of *abstraction*: Given a variable x and an expression t (possibly containing x), there exists a function f such that $f\,x = t$ which can be generalized by simply writing as a mapping of the form $x \longmapsto t$. When working in the context of the λ-calculus, we denote this function by $\lambda x.t$. For instance, we may write

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

to denote the function

$$a \longmapsto \lambda y.\lambda z.(a + y + z)$$

As will be seen in the proceeding chapter, representing functions in this manner allows us to more easily identify and work with functions when such abstractions are nested in an expression.

The proof-theoretical aspects of the λ-calculus arise when we require that every expression come equipped with a type X, as the λ-calculus consists of axioms and inference rules for deriving valid typing λ-expressions, in which we write $t : X$ to denote that the expression t is of type X. Note that from a purely mathematical perspective, we can simply consider types as sets so $t : X$ could just as well be thought of as meaning that $t \in X$. Although there are untyped formulations of the λ-calculus, for the purposes of this thesis, we will only consider those species of λ-calculi in which every expression (term) comes equipped with a corresponding type.

Although the intuitionistic propositional calculus and the typed λ-calculus are *prima facie* dramatically distinct systems of proof-theoretical calculi, they are in fact very closely related at a fundamental level. The Curry-Howard isomorphism, due to Haskell Curry and William Howard, is one of the most celebrated series of results in proof theory and represents a paradigm between axiomatic propositional calculi, natural deduction calculi, sequent calculi, and typed λ-calculi. The beginning fragments of this correspondence were first noticed by Curry (1934) who observed a strong connection between types of combinators and axiomatic implicational intuitionistic logic. It was then later noticed by Curry (1958) that there is a strong correspondence between the typed fragment of combinatory logic and Hilbert style axiomatic calculi. Most importantly (at least for our current purposes) it was then noticed by Howard (1969) that there is a strong correspondence between the intuitionistic propositional calculus and the typed λ-calculus in the sense that proofs in the former can be interpreted by proofs in the latter.

Category theory, on the other hand, is much more algebraic in nature. The theory of categories was originally formulated by Samuel Eilenberg and Saunders MacLane in their influential article "General Theory of Natural Equivalences"(1945). In this work, they sought to develop an abstract algebraic framework constructed by some basic algebraic operations and axioms, sufficient in characterizing the notion of a structure preserving transformation between various species of mathematical structures, such as vector spaces and their linear transformations, topological spaces and their continuous functions, or groups and their group homomorphisms. The primary concern which motivated Eilenberg and MacLane to characterize transformations between structures with this property in the first place was that of the inability of their current mathematical techniques in offering a "natural" understanding of the notion of an isomorphism between such structures; natural in the sense that the isomorphisms are uniquely determined.[2] Their resulting notion of a natural or "adjoint" equivalence between such structures turned out to be dependent upon the notion of a natural transformation, which turned out to be dependent upon that of a functor, which was lastly turned out to be dependent upon the notion of a

[2]For example, they recognized that the notion of a category allowed them to construct a truly natural isomorphism between vector spaces and their linear transformations in the sense that the resulting isomorphisms are independent of the particular base points in the vector spaces one chooses.

category.

Being a field of abstract algebra in its own right, the study of category theory can be roughly characterized by the study of abstract algebras that arise from a system of abstract functions. One of the primary advantages that is afforded to us by category theory is directly seen through its ability to describe and characterize mathematical structures and properties of mathematical structures at an extremely generalized level. As a consequence, categorical methods typically involve abstracting away much of the data of some mathematical structure down to its basic abstract algebraic (i.e., category-theoretical) components. This allows us in turn to compare drastically distinct species of mathematical structures with one another and even posit the notion of an equivalence or isomorphism between such structures, which we otherwise would be unable to compare. Aside from category theory being uniquely equipped to offer a precise notion of an equivalence of structures at an extremely generalized level, another useful methodological advantage that is offered to us by category theory is its ability to “categorify” or translate a statement from some other area in mathematics to a reformulated statement within the language of category theory. The extremely powerful nature of the category-theoretical apparatus then makes it so that such “categorified” statements can either be more easily demonstrated, understood in different ways, or understood in relation to other structures.

Although the intuitionistic propositional calculus and the typed λ-calculus (on the logical side of things) appear to be quite distinct species of mathematical systems with respect to category theory (on the algebraic side of things), there are in fact many deep connections that can be made between them, which will be the primary subject matter of this thesis. Categorical logic, roughly speaking, is an approach to category theory which places emphasis on its applications to logic. We will concern ourselves with the applications between Cartesian closed categories and proof-theoretical calculi previously discussed. This particular form of interaction that occurs between Cartesian closed categories and both the intuitionistic propositional calculus along with the typed λ-calculus was developed by William Lawvere (1963/1969a/1969b), Joachim Lambek (1968/1969/1972/1974), and then in great detail by Lambek and Scott (1986). We will examine the interaction that occurs between these structures via the use of algebraic theories, which enjoy the property of offering a categorical semantics for the typed λ-calculus. This particular technique was formulated by Lawvere (1963) in his influential "Functorial Semantics of Algebraic Theories".

The intended contribution of this work is to give a clear picture of the mathematics described above in such a way that it can be appreciated by those researchers in logic who are not necessarily acquainted with the the algebraic techniques of category theory. Whereas many texts in category theory, and even categorical logic specifically, often de-emphasize the logical dimensions of the subject, or pass over the subtle categorical dimensions, I aim to give a more explicit treatment thereby clarifying the notion that the perspective of the Cartesian closed category-theorist is essentially equivalent to the perspectives of the type theorist and intuitionistic logician.

Chapter 1

Proof-Theoretical Foundations for Intuitionistic Logic

The main purpose of this chapter is to equip the reader with the necessary proof-theoretical apparatus required in order to come to grips with the positive fragment of the intuitionistic propositional calculus, the typed λ-calculus, along with the Curry-Howard isomorphism, which relates the former and latter species of proof-theoretical calculi.

1.1 Intuitionistic Propositional Calculus

We begin by constructing the underlying language L of the intuitionistic propositional calculus in the following manner: Let $Var = \{p_1, p_2, p_3, \ldots,\}$ be a countably infinite set of propositional variables. We build formulas from the set of propositional variables by connectives for conjunction $\wedge$, disjunction $\vee$, and implication $\Rightarrow$. We then make use of the variables ϕ, ψ, $\omega, \ldots$ by taking them as ranging over arbitrary formulas in L.

Definition 1.1. The *formulas* of L are defined inductively as follows:

1. Every propositional variable p_k for $k \in \mathbb{Z}^+$ is a formula
2. The constants $\bot$ (falsity) and $\top$ (truth) are formulas
3. If ϕ and ψ are formulas, then $\phi \wedge \psi$ is a formula
4. If ϕ and ψ are formulas, then $\phi \vee \psi$ is a formula
5. If ϕ and ψ are formulas, then $\phi \Rightarrow \psi$ is a formula
6. Nothing else is a formula

Given the notion of a well formed formula in the language L, we use Γ to denote a finite set of formulas $\Gamma = \{\phi_1, \phi_2, \phi_3, \ldots\}$ and then consider *judgments* of the form $\Gamma \vdash \phi$ which denotes the existence of a proof that starts from the assumption of the finite set of formulas Γ and results in the formula ϕ. When a formula ϕ is provable from the empty set of assumptions, we write $\vdash \phi$ to mean that ϕ is a *theorem* in L. We define proofs (or derivations) of judgments according to the following definition.

Definition 1.2. Let $\Gamma \vdash \phi$ be a judgment. A *proof* (or *derivation*) of $\Gamma \vdash \phi$ is a finite tree T of judgments satisfying the following conditions:

1. The label of the root node (i.e., the bottom-most node) of T is $\Gamma \vdash \phi$
2. The label of every leaf node (i.e., the upper-most node) of T is an axiom, namely a judgment of the form $\phi \vdash \phi$
3. The label of any node which descends from the label of a leaf node of T is a result of applying finitely many axioms and inference to the labels of the leaf nodes in T

We then acquire the intuitionistic propositional calculus by starting with the language L and requiring that judgments that are built from L satisfy a single axiom and be closed under a collection of inference rules for $\top$, $\bot$ $\wedge$, $\vee$, and $\Rightarrow$, as stated below.

Definition 1.3. The *intuitionistic propositional calculus* is characterized by the following axiom and collection of inference rules:

- There is exactly one axiom:

$$\phi \vdash \phi$$

- There is an introduction rule for $\top$ defined by:

$$\frac{}{\Gamma \vdash \top}\ \top I$$

- There is one elimination rule for $\bot$ defined by:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash \phi}\ \bot\mathrm{E}$$

- There is one introduction rule for $\wedge$ defined by:

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}\ \wedge I_1$$

- There are two elimination rules for $\wedge$ defined by:

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}\ \wedge E_1 \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}\ \wedge E_2$$

- There are two introduction rules for $\vee$ defined by:

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi}\ \vee I_1 \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi}\ \vee I_2$$

- There is one elimination rule for $\vee$ defined by:

$$\frac{\Gamma \vdash \phi \vee \psi \qquad \Gamma, \phi \vdash \omega \qquad \Gamma, \psi \vdash \omega}{\Gamma \vdash \omega} \vee E$$

- There is an introduction and an elimination rule for $\Rightarrow$ defined by:

$$\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \Rightarrow \psi} \Rightarrow I \qquad\qquad \frac{\Gamma \vdash \phi \Rightarrow \psi \qquad \Gamma \vdash \phi}{\Gamma \vdash \psi} \Rightarrow E$$

For the remainder of this chapter, we will confine ourselves to the positive fragment of the intuitionistic propositional calculus including $\top$, $\wedge$, and $\Rightarrow$, which is often referred to as *positive minimal logic.* We will later adopt a slightly more precise view of proofs within the positive intuitionistic propositional calculus by keeping track of which assumptions are discharged and where. We will however first introduce the typed λ-calculus as it in fact motivates making this subtle refinement.

1.2 Typed λ-Calculus

As earlier mentioned, unlike the intuitionistic propositional calculus which is a formal calculus of propositions, the typed λ-calculus is a formal calculus of functions. We proceed by giving a description of the typed λ-calculus by defining its constituent collection of types, terms, and equations in turn.

Definition 1.4. The collection *Type*(Λ) of types belonging to the typed λ-calculus consists of the following data:

1. Basic types: We express that a type X is among the basic types by

$$\frac{}{X \ \mathit{type}}$$

which means that X is a type. There is always a *unit type* 1 so

$$\frac{}{1\ \mathit{type}}$$

2. Product types $X \times Y$ and function types $X \longrightarrow Y$ constructed by the following type-forming operations

$$\frac{X\ \mathit{type} \qquad Y\ \mathit{type}}{X \times Y\ \mathit{type}} \qquad \frac{X\ \mathit{type} \qquad Y\ \mathit{type}}{X \longrightarrow Y\ \mathit{type}}$$

   In particular, the set of all types are constructed from the basic types via product and function type-forming operations. Note that function types are right-associative in the sense that for arbitrary types X, Y, Z,

$$X \longrightarrow Y \longrightarrow Z \equiv X \longrightarrow (Y \longrightarrow Z)$$

Now that we have established the collection of typed belonging to the typed λ-calculus, we proceed by considering its collection of terms.

Definition 1.5. The collection $\mathit{Term}(\Lambda)$ of *terms* belonging to the typed λ-calculus is then constructed from a countable set of *variables* $\{x, y, z, \ldots\}$ along with a set of *basic constants* by the following term-forming operations:

$$t := [\mathit{Var}] \mid [\mathit{Const}] \mid \star \mid \langle t, u \rangle \mid \pi_1 t \mid \pi_2 t \mid \lambda x : X.t$$

defined recursively by the following clauses:

1. any variable x is a term
2. every basic constant a is a term
3. the constant $\star$, i.e., the *unit*, is a term

4. if t and u are terms, then their *pariring* $\langle t, u\rangle$ is a term

5. if t is a term, then so are $\pi_1 t$ and $\pi_2 t$

6. if t and u are terms, then so is their *application* $t\,u$

7. given a variable x, a type X, and a term t, the *λ-abstraction* $\lambda x : X.t$ is a term

Note that application is left-associative in the sense that for arbitrary terms t, u, v, we have $t\,u\,v = (t\,u)v$.

Definition 1.6. Given any term t, we compute the free variables *Freevar*(t) according to the following clauses:

- $\mathit{Freevar}(x) = \{x\}$ if $x \in \mathit{Var}$
- $\mathit{Freevar}(a) = \emptyset$ if $a \in \mathit{Const}$
- $\mathit{Freevar}(\langle t, u\rangle) = \mathit{Freevar}(t) \cup \mathit{Freevar}(u)$
- $\mathit{Freevar}(\pi_1 t) = \mathit{Freevar}(t)$
- $\mathit{Freevar}(\pi_2 t) = \mathit{Freevar}(t)$
- $\mathit{Freevar}(t\,u) = \mathit{Freevar}(t) \cup \mathit{Freevar}(u)$
- $\mathit{Freevar}(\lambda x.t) = \mathit{Freevar}(t) \setminus \{x\}$

Given terms t, u and a variable x, we can build new terms of the form $t[u/x]$ by substituting u for x in t. Note that substitution gives us rules of replacement for only the free occurrences of x. By renaming those variables that are bound in t in such a way that they are disjoint from those variables that freely occur in t, we arrive at the following definition:

Definition 1.7. The substitution rules for the typed λ-calculus are given by the following equations:

1. $x[u/x] = u$
2. $y[u/x] = y$ if $x \neq y$
3. $a[u/x] = a$ if $a \in Const$
4. $\langle s, t\rangle[u/x] = \langle s[u/x], t[u/x]\rangle$
5. $(\pi_1 t)[u/x] = \pi_1(t[u/x])$
6. $(\pi_2 t)[u/x] = \pi_2(t[u/x])$
7. $(s\,t)[u/x] = (s[u/x])(t[u/x])$
8. $(\lambda y : X.t)[u/x] = \begin{cases} \lambda y : X.(t[u/x]) & \text{if } x \neq y, y \notin \textit{Freevar(u)} \\ \lambda y : X.t & \text{otherwise} \end{cases}$

Recall in the last section, we established the notion of a judgment (provability) relation between a set of formulas Γ and a single formula ϕ which is defined when ϕ is the result of applying finitely many axioms and inference rules to the the formulas contained in Γ. We wish to define a similar relation of the types and terms within the typed λ-calculus.

Definition 1.8. Let $x_1, \ldots, x_n$ be a finite sequence of distinct variables and let $X_1, \ldots, X_n$ be a finite sequence of types. A *typing context* is of the form

$$x_1 : X_1 \ldots, x_n : X_n$$

which we denote by $\ldots, \Gamma, \Delta, \Omega, \ldots$. Then, a *typing judgment* is of the form

$$\Gamma = x_1 : X_1, \ldots, x_n : X_n \vdash t : X$$

which we read as "the term t is of type X in the typing context Γ".

We then require that the collection of valid typing judgments be closed under the following axioms and inference rules:

Definition 1.9. The collection of *valid typing judgments* in the typed λ-calculus are characterized by the following axioms and inference rules:

1. Every basic constant a is of a uniquely determined type X,

$$\frac{}{\Gamma \vdash a : X}$$

2. Types for variables are entirely determined by the context:

$$\frac{}{x_1 : X_1, \ldots, x_k : X_k, \ldots, x_n : X_n \vdash x_k : X_k}\ (1 \leq k \leq n)$$

3. The (unit) constant $\star$ is of type 1

$$\frac{}{\Gamma \vdash \star : 1}$$

4. Typing rules for the projections $\pi_1 t$ and $\pi_2 t$ are given by

$$\frac{\Gamma \vdash t : X \times Y}{\Gamma \vdash \pi_1 t : X} \qquad \frac{\Gamma \vdash t : X \times Y}{\Gamma \vdash \pi_2 t : Y}$$

5. The typing rule for the pairing operator $\langle t, u \rangle$ is given by

$$\frac{\Gamma \vdash t : X \qquad \Gamma \vdash u : Y}{\Gamma \vdash \langle t, u \rangle : X \times Y}$$

6. The typing rule for application is given by:

$$\frac{\Gamma \vdash t : X \longrightarrow Y \qquad \Gamma \vdash u : X}{\Gamma \vdash t\, u : Y}$$

7. The typing rule for λ-abstraction is given by:

$$\frac{\Gamma, x : X \vdash t : Y}{\Gamma \vdash (\lambda x : X.t) : X \longrightarrow Y}$$

The last set of conditions necessary for giving a complete characterization of the typed λ-calculus is to make explicit the closure rules for equations between terms. Note that given terms t and u of type X in context Γ as in $\Gamma \vdash t : X$ and $\Gamma \vdash u : X$, we denote the judgment that t and u are equal by

$$\Gamma \vdash t = u : X$$

Here, the terms t and u are necessarily of the same type.

Definition 1.10. The collection $Eq(\Lambda)$ of *equations* between terms belonging to the typed λ-calculus are closed under the following rules:

1. Equality $=$ is defined as an equivalence relation by

$$\frac{}{\Gamma \vdash t = t : X}\text{ Ref.} \qquad \frac{\Gamma \vdash t = u : X}{\Gamma \vdash u = t : X}\text{ Sym.}$$

$$\frac{\Gamma \vdash t = u : X \qquad \Gamma \vdash u = v : X}{\Gamma \vdash t = v : X}\text{ Trans.}$$

2. A rule for weakening the context:

$$\frac{\Gamma \vdash t = u : X}{\Gamma, x : Y \vdash t = u : X}$$

3. A rule for the unit type:

$$\frac{\Gamma \vdash t : 1}{\Gamma \vdash t = \star : 1}$$

4. Equations for product types are given by:

$$\frac{\Gamma \vdash t = u : X \qquad \Gamma \vdash v = s : Y}{\Gamma \vdash \langle t, v \rangle = \langle u, s \rangle : X \times Y}$$

$$\frac{\Gamma \vdash t = u : X \times Y}{\Gamma \vdash \pi_1 t = \pi_1 u : X} \qquad \frac{\Gamma \vdash t = u : X \times Y}{\Gamma \vdash \pi_2 t = \pi_2 u : X}$$

$$\frac{}{\Gamma \vdash \pi_1 \langle t, u \rangle = t : X} \qquad \frac{}{\Gamma \vdash \pi_2 \langle t, u \rangle = u : X}$$

$$\frac{}{\Gamma \vdash t = \langle \pi_1 t, \pi_2 t \rangle : X \times Y}$$

5. Equations for function types are given by:

$$\frac{\Gamma \vdash t = u : X \longrightarrow Y \qquad \Gamma \vdash v = s : X}{\Gamma \vdash t v = u s : Y}$$

$$\frac{\Gamma, x; X \vdash t = u : Y}{\Gamma \vdash (\lambda x : X.t) = (\lambda x : X.u) : X \longrightarrow Y}$$

$$\frac{}{\Gamma \vdash (\lambda x : X.t)u = t[u/x] : X} \ \beta\text{-rule}$$

$$\frac{}{\Gamma \vdash \lambda x : X.(t\,x) = t : X \longrightarrow Y} \ \text{if } x \notin \mathit{Freevar(t)} - (\eta)\text{-rule}$$

1.3 Remarks on the Curry-Howard Isomorphism

As mentioned at the outset of this work, the Curry-Howard isomorphism is not not a single unified theorem in proof theory, but rather represents a paradigm between logical and computational calculi. The original aspect of this paradigm was noticed by Curry who showed that strong connections can be made between Hilbert-style axiomatic propositional calculus and combinatory logic, in his "Functionality in Combinatory Logic" (1934). Although the Curry-Howard isomorphism has since been shown to extend to many distinct species of logical and computational calculi, we will concern ourselves with the ways in which it relates the respective calculi constructed in the previous two sections of this chapter; namely, that of the positive intuitionistic propositional calculus and the typed λ-calculus. As this is merely a background result with respect to the purposes of this thesis, we will give only a brief sketch of this particular fragment of the correspondence.

William Howard was the first to extend this logical-computational paradigm to that of the typed λ-calculus and the intuitionistic propositional calculus in his influential manuscript, "The Formulae-as-Types notion of Construction" (Howard (1969)). Howard's key observation was that the introduction and elimination rules within the positive intuitionistic propositional calculus correspond to the right-hand side of the typing judgments for term-forming operators within the typed λ-calculus, as seen below:

IPC^+	λ-Calculus
$\phi \vdash \phi$	$x : X \vdash x : X$
$\dfrac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \wedge I_1$	$\dfrac{\Gamma \vdash t : X \qquad \Gamma \vdash u : Y}{\Gamma \vdash \langle t, u \rangle : X \times Y}$
$\dfrac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \wedge E_1$	$\dfrac{\Gamma \vdash t : X \times Y}{\Gamma \vdash \pi_1 t : X}$
$\dfrac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi} \wedge E_2$	$\dfrac{\Gamma \vdash t : X \times Y}{\Gamma \vdash \pi_2 t : Y}$
$\dfrac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \Rightarrow \psi} \Rightarrow I$	$\dfrac{\Gamma, x : X \vdash t : Y}{\Gamma \vdash (\lambda x : X . t) : X \longrightarrow Y}$
$\dfrac{\Gamma \vdash \phi \Rightarrow \psi \qquad \Gamma \vdash \phi}{\Gamma \vdash \psi} \Rightarrow E$	$\dfrac{\Gamma \vdash t : X \longrightarrow Y \qquad \Gamma \vdash u : X}{\Gamma \vdash t\, u : Y}$

Hence, Howard's fundamental observation was that truth $\top$ within the intuitionistic propositional calculus is interpreted as the unit type 1 within the typed λ-calculus, the conjunction of formulas $\phi \wedge \psi$ is interpreted as the product of two types $X \times Y$, and an implication $\phi \Rightarrow \psi$ between two formulas is interpreted as the function type $X \longrightarrow Y$.

The second observation that is made by the Curry-Howard isomorphism then extends this observation by noticing that applying the normalization procedure to proofs within the positive intuitionistic propositional calculus corresponds to the reduction of terms within the typed λ-calculus. Note that a derivation within the positive intuitionistic propositional calculus (and more generally, any propositional calculus) is considered to be in *normal form* provided there is no introduction rule for some connective that is immediately proceeded by an elimination rule for that same connective (i.e., the derivation contains no detours). Derivations within a propositional calculus are then said to be *normalizable* provided each derivation can be transformed into a derivation in normal form. The normalization procedure for derivations within the positive intuitionistic propositional calculus is therefore carried out by removing such detours, as in the following:

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \Gamma \vdash \phi\end{array} \qquad \begin{array}{c}\pi'\\ \Gamma \vdash \psi\end{array}}{\Gamma \vdash \phi \wedge \psi}\wedge I}{\Gamma \vdash \phi}\wedge E_1 \quad \longrightarrow \quad \begin{array}{c}\pi\\ \Gamma \vdash \phi\end{array}$$

or as in:

$$\dfrac{\dfrac{\begin{array}{c}\pi\\ \Gamma \vdash \phi\end{array} \qquad \begin{array}{c}\pi'\\ \Gamma \vdash \psi\end{array}}{\Gamma \vdash \phi \wedge \psi}\wedge I}{\Gamma \vdash \psi}\wedge E_2 \quad \longrightarrow \quad \begin{array}{c}\pi'\\ \Gamma \vdash \psi\end{array}$$

Note that within the above examples, whether the detour normalizes to π or π' is simply a matter of which elimination rule is applied to obtain the result. Sometimes however, the removal of detours needs to only be applied to sub-derivations within some larger derivation as in the following case:

$$\dfrac{\dfrac{\begin{matrix}\pi\\ \Gamma,\phi\vdash\psi\end{matrix}}{\Gamma\vdash\phi\Rightarrow\psi}\Rightarrow I \qquad \begin{matrix}\pi'\\ \Gamma\vdash\phi\end{matrix}}{\Gamma\vdash\psi} \quad\longrightarrow\quad \begin{matrix}\pi[\pi'/\phi]\\ \Gamma\vdash\psi\end{matrix}$$

where $\pi[\pi'/\phi]$ is the result of replacing every initial judgment of the form $\phi \vdash \phi$ within the sub-derivation π' whose last line is $\Gamma \vdash \phi$, in the sub-derivation π. This can be easily seen to correspond exactly to the reduction rules for terms within the typed λ-calculus listed below:

$$\pi_1\langle t,u\rangle = t, \quad \pi_2\langle t,u\rangle = u, \quad (\lambda x.t)u = t[u/x]$$

by the following corresponding examples, respectively:

$$\dfrac{\dfrac{\begin{matrix}\pi\\ \Gamma\vdash t:X\end{matrix} \qquad \begin{matrix}\pi'\\ \Gamma\vdash u:Y\end{matrix}}{\Gamma\vdash\langle t,u\rangle:X\times Y}}{\Gamma\vdash\pi_1\langle t,u\rangle:X} \quad\longrightarrow\quad \begin{matrix}\pi\\ \Gamma\vdash t:X\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\pi\\ \Gamma\vdash t:X\end{matrix} \qquad \begin{matrix}\pi'\\ \Gamma\vdash u:Y\end{matrix}}{\Gamma\vdash\langle t,u\rangle:X\times Y}}{\Gamma\vdash\pi_2\langle t,u\rangle:Y} \quad\longrightarrow\quad \begin{matrix}\pi'\\ \Gamma\vdash u:Y\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\pi\\ \Gamma,x:X\vdash t:Y\end{matrix}}{\Gamma\vdash\lambda^X.t:X\longrightarrow Y} \qquad \begin{matrix}\pi'\\ \Gamma\vdash u:X\end{matrix}}{\Gamma\vdash(\lambda^X.t)u:Y} \quad\longrightarrow\quad \begin{matrix}\pi[\pi'/X]\\ \Gamma\vdash t[u/x]:Y\end{matrix}$$

In summary, we have observed that: 1.) the introduction and elimination rules within the positive intuitionistic propositional calculus correspond to the right-hand side of typing judgments for term-forming operators within the typed λ-calculus and, 2.) applying the normalization procedure to derivations within the positive intuitionistic propositional calculus corresponds to the reduction of terms within the typed λ-calculus. These are the key fundamental observations involved in the fragment Curry-Howard isomorphism which is of interest to us for the purposes of this work. For more details on the Curry-Howard isomorphism, please see Sorensen and Urzyczyn (2006).

Chapter 2

Foundations for Cartesian Closed Category Theory

The main purpose of this chapter is to equip the reader with a sufficient amount of category-theoretical apparatus required in order to come to grips with the principal theorem of this thesis found in chapter 4. This will most importantly involve making explicit the notions of a Cartesian closed category and that of an equivalence of categories. For a more in-depth treatment of pure category theory, please see MacLane (1971), Barr and Wells (1995), and Awodey (2010).

2.1 The Definition of a Category

Definition 2.1. A *category* $\mathscr{C}$ is any structure consisting of the following data:

- A collection $Ob(\mathscr{C})$ of objects $\ldots, X, Y, Z, \ldots$

- A collection $Map(\mathscr{C})$ of mappings $\ldots, f, g, h, \ldots$ in which for every map $f \in Map(\mathscr{C})$, there exists a pair of objects $(X, Y) \in Ob(\mathscr{C})$ with

$$dom(f) = X,\ cod(f) = Y$$

which we denote through either of the following conventions

$$f : X \longrightarrow Y \quad \text{or} \quad X \xrightarrow{f} Y$$

Moreover, we require that this data satisfy the following conditions

- For every object X, there exists an identity map

$$1_X : X \longrightarrow X$$

- Two maps f and g are composable if $cod(f) = dom(g)$. Then, for every composable pair of maps (f, g) such that $f : X \longrightarrow Y$ and $g : Y \longrightarrow Z$, there exists a composition map given by

$$gf : X \longrightarrow Z$$

Lastly, we require that the entire composition structure of $\mathscr{C}$ satisfy the following axioms:

- The composition operation is unitial in the sense that for every map $f : X \longrightarrow Y$, the following equation is satisfied

$$f 1_X = f = 1_Y f$$

- The composition operation is associative in the sense that for every composable triple of maps (f, g, h) such that $f : W \longrightarrow X$, $g : X \longrightarrow Y$, and $h : Y \longrightarrow Z$, the following equation is satisfied

$$h(gf) = (hg)f$$

A nice conceptual advantage that is offered to us by the language of category theory is that it adopts a slight geometric character by virtue of its equations being expressible as commutative diagrams. For example, we can represent the composition operation defined over any pair of composable maps in a category by the following commutative diagram

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ & {\scriptstyle gf}\searrow & \downarrow{\scriptstyle g} \\ & & Z \end{array}$$

The commutativity of the above diagram entails that applying the map f to X and then g to Y is the same as applying their composition gf to X. It is in this sense in which commutative diagrams represent an equality of mappings in a category. Therefore, we can represent the equation which guarantees that composition is unitial by

$$1_X \circlearrowright X \xrightarrow{f} Y \circlearrowleft 1_Y$$

as this implies that the map defined by applying the identity map 1_X to X and then composing with the map f, as well as the map defined by first applying f to X and then composing the identity map 1_Y, as well as the map simply defined by applying f to X are equal. Lastly, we can represent the equation that guarantees that composition is associative by

$$\begin{array}{ccccc} W & \xrightarrow{f} & X & & \\ & {\scriptstyle gf}\searrow & \downarrow{\scriptstyle g} & \searrow{\scriptstyle hg} & \\ & & Y & \xrightarrow[h]{} & Z \end{array}$$

as this implies that applying the map defined by composing the composition hg after f is equal to the map defined by composing h after the composition gf.

2.2 Examples of Categories

With the general notion of a category in mind, we proceed by discussing some examples of commonplace place mathematical structures and objects which give rise to categories.

Example 2.2. The category **SET** of sets consists of:

- Objects: Given by the class of sets $\dots, X, Y, Z, \dots$
- Maps: Given by the class of set theoretical-functions $\dots, f, g, h, \dots$
- Given any set X, the identity function $1_X : X \longrightarrow X$ is defined by $1_X(x) = x$ for $x \in X$
- Given any composable pair of functions $f : X \longrightarrow Y$, $g : Y \longrightarrow Z$, their composition $gf : X \longrightarrow Z$ is defined by $g(f(x))$ for $x \in X$.

There are also categories that arise not from a system of functions on sets, but rather, from a system of functions on structured sets.

Example 2.3. The category **TOP** of topological spaces consists of:

- Objects: Given by a class of topological spaces $(X, \mathscr{O})$ in which X is a non-empty set and $\mathscr{O} \subseteq X$ is a family of open subsets of X defined by the following conditions:
  1. $\emptyset, X \in \mathscr{O}$,
  2. if $U_\alpha \in \mathscr{O}$, then $\bigcup_{\alpha \in A} U_\alpha \in \mathscr{O}$,
  3. if $U_k \in \mathscr{O}$, then $\bigcap_{k=1}^{n} U_k \in \mathscr{O}$

- Maps: Given by a class of continuous functions $f : (X, \mathcal{O}) \longrightarrow (Y, \mathcal{O}')$ defined by the condition that for every open subset $U \in \mathcal{O}'$, the inverse function $f^{-1}(U) \in \mathcal{O}$

- Given any topological space $(X, \mathcal{O})$, there exists a continuous identity function $1_{(X,\mathcal{O})} : (X, \mathcal{O}) \longrightarrow (X, \mathcal{O})$

- Given any composable pair of continuous functions $f : (X, \mathcal{O}) \longrightarrow (Y, \mathcal{O}')$ and $g : (Y, \mathcal{O}') \longrightarrow (Z, \mathcal{O}'')$ their composition is given by the continuous function $gf : (X, \mathcal{O}) \longrightarrow (Z, \mathcal{O}'')$.

Note that the maps in a category needn't even necessarily be given by functions, as in the following:

Example 2.4. The category **REL** of relations consists of:

- Objects: Given by a class of sets $\dots, X, Y, Z, \dots$

- Maps: Given by a class of relations $R \subseteq X \times Y$ between sets

- Identity is given by the diagonal relation defined by

$$1_X = \{(x, x) \mid x \in X\}$$

- The composition operation is given by the usual composition of relations defined by $SR = \{(x, z) \mid \exists y.(x, y) \in R, (y, z) \in S\}$

Here is another example of a category whose maps are not given by set-theoretical functions, and is one that will offer us our first glimpse at how the viewpoints of category theory and logic intersect.

Example 2.5. The category **TL-ALG** is given by a Tarski-Lindenbaum algebra and consists of:

- Objects: Given by equivalence classes $[\phi]$ of formulas ϕ defined inductively by the language L as in definition 1.1, which are determined uniquely up to provable equivalence, as in $[\phi]=[\psi] \Longleftrightarrow \phi \dashv\vdash \psi$
- Maps: Given by a partial ordering $\leq$ over the equivalence classes of formulas in L defined by $[\phi] \leq [\psi] \Longleftrightarrow \phi \vdash \psi$
- Identity: Given by the reflexivity of the provability relation $\vdash$ in that for any formula ϕ, $\phi \vdash \phi$, so $[\phi] \leq [\phi]$
- Composition: Given by the transitivity of the provability relation $\vdash$ in that for any formulas ϕ, ψ, and ω such that $\phi \vdash \psi$ and $\psi \vdash \omega$, there exists a proof $\phi \vdash \omega$, so if $[\phi] \leq [\psi]$ and $[\psi] \leq [\omega]$, then $[\phi] \leq [\omega]$.

Note that the category **LT-ALG** is more precisely an example of what is known as a *poset category*, in which there exists at most one map between any two objects. Although there exists countably many proofs from ϕ to ψ if $\phi \vdash \psi$, the satisfaction of this relation merely defines when a map exists between two objects, i.e., when the relation $\leq$ is defined between the equivelance classes of ϕ and ψ. Although this list doesn't come close to including all those species of mathematical structures and systems that give rise to categories, it should give the reader somewhat of a sense of what the objects and maps in a category may be given by in practice. For the purpose of this chapter, we will typically refer back to the category **SET** and **LT-ALG** when attempting to give a concrete example of some abstract category-theoretical definition or construction.

Example 2.6. The last species of categories worth mentioning are that of opposite (or dual categories). Given a category $\mathscr{C}$, the opposite category

$\mathscr{C}^{op}$ is a result of formally turning around every map in $\mathscr{C}$. Hence, given a category $\mathscr{C}$, the opposite category $\mathscr{C}^{op}$ consists of:

- Objects: Same as those in $\mathscr{C}$
- Maps: For every map $f : X \longrightarrow Y$ in $\mathscr{C}$, there is a map $\bar{f} : \bar{Y} \longrightarrow \bar{Z}$ in $\mathscr{C}^{op}$
- Identity: For every identity map $1_X : X \longrightarrow X$ in $\mathscr{C}$, there is a map $\bar{1}_X : \bar{X} \longrightarrow \bar{X}$ in $\mathscr{C}^{op}$
- Composition: for every composition of maps $gf : X \longrightarrow Z$ in $\mathscr{C}$ given explicitly by the following commutative diagram

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ & \searrow^{gf} & \downarrow g \\ & & Z \end{array}$$

there is a composition of maps $\bar{f}\bar{g} : \bar{Z} \longrightarrow \bar{X}$ in $\mathscr{C}^{op}$ given by the following commutative diagram

$$\begin{array}{ccc} \bar{Z} & \xrightarrow{\bar{g}} & \bar{Y} \\ & \searrow^{\bar{f}\bar{g}} & \downarrow \bar{f} \\ & & \bar{X} \end{array}$$

Hence, given any statement Σ in the language of category theory, we can form its dual statement $\bar{\Sigma}$ by replacing $dom(f)$ by $cod(f)$, $cod(f)$ by $dom(f)$, and gf by fg for all maps f and g. More generally, we have the following *categorical principle of duality*:

Proposition 2.7. For any statement Σ in the language of category theory, if Σ follows from the axioms of categories, then so does its dual statement $\bar{\Sigma}$, i.e., if $\mathscr{C} \vdash \Sigma$, then $\mathscr{C} \vdash \bar{\Sigma}$.

2.3 Properties of Maps

The purpose of this section is to establish the notion of an isomorphism between two objects in a category, which at its core, is a purely category-theoretically definable mapping property.

Definition 2.8. Let $\mathscr{C}$ be a category and let $f : X \longrightarrow Y$ be a map in $\mathscr{C}$. Then, f is a *monomorphism* if for any maps $g_1, g_2 : W \longrightarrow X$,

$$f g_1 = f g_2 \Longrightarrow g_1 = g_2$$

making the following diagram commute

$$W \underset{g_2}{\overset{g_1}{\rightrightarrows}} X \xrightarrow{f} Y$$

Moreover, f is an *epimorphism* if for any maps $h_1, h_2 : Y \longrightarrow Z$,

$$h_1 f = h_2 f \Longrightarrow h_1 = h_2$$

making the following diagram commute

$$X \xrightarrow{f} Y \underset{h_2}{\overset{h_1}{\rightrightarrows}} Z$$

Definition 2.9. Let $\mathscr{C}$ be a category and let $f : X \longrightarrow Y$ be a map in $\mathscr{C}$. Then, f is a *section* if there exists a map $f' : Y \longrightarrow X$ such that $f'f = 1_X$ making the following triangle commute

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ & \searrow^{1_X} & \downarrow f' \\ & & X \end{array}$$

Moreover, a map $f : X \longrightarrow Y$ in $\mathscr{C}$ is a *retraction* if there exists a map $f' : Y \longrightarrow X$ such that $f f' = 1_Y$ making the following triangle commute

$$\begin{array}{ccc} Y & \xrightarrow{f'} & X \\ & \searrow^{1_Y} & \downarrow f \\ & & Y \end{array}$$

With the concept of a section and a retraction in mind, we arrive the notion of an isomorphism between objects in a category.

Definition 2.10. Let $\mathscr{C}$ be a category and let $f : X \longrightarrow Y$ be a map in $\mathscr{C}$. Then, f is an *isomorphism* if f is both a section and a retraction. That is, if $f'f = 1_X$ and $ff' = 1_Y$ for some map $f' : Y \longrightarrow X$ in $\mathscr{C}$

Note that we typically denote the fact that $f : X \longrightarrow Y$ is an isomorphism by writing $X \cong Y$. Furthermore, note that we can alternatively characterize the notion of an isomorphism between objects in a category as follows:

Lemma 2.11. Let $\mathscr{C}$ be a category and let $f \in \mathscr{C}$ be an arbitrary map. Then, the following statements are equivalent:

1. f is an isomorphism
2. f is a monomorphic retraction
3. f is an epimorphic section

Proof. Note that 1 implies both 2 and 3 since if $f : X \longrightarrow Y$ is an isomorphism, then f is both a section and a retraction by definition 2.10. Moreover, since every section is a monomorphism and every retraction is an epimorphism, f is both a monomorphic retraction and an epimorphic section. Since 2 and 3 are dualized statements, it suffices to show that 2 implies 1. Since f is a section, it has a right inverse f' so we know that $ff'f = f$, but since f is a monomorphism, $f'f = 1_X$ making f a section. Then since f is both a section and a retraction, f is an isomorphism. □

Remark 2.12. Note that in the category **SET**, the monomorphisms are given exactly by the injective (one-one) functions, i.e., the class of all set-theoretical functions $f : X \longrightarrow Y$ such that for all $x \in X$ and $y \in Y$, if $f(x) = f(y)$, then $x = y$. Moreover, the epimorphisms are given exactly by the surjective (onto) functions, i.e., the class of all set-theoretical functions $f : X \longrightarrow Y$ in which for every $y \in Y$, there exists $x \in X$ such that $f(x) = y$. Lastly, in the category **SET**, every monomorphism is a section and every epimorphism is a retraction, so the isomorphisms in **SET** are given exactly by the bijective (one-one and onto) functions, i.e., the class of all set-theoretical functions that are both injective and surjective.

2.4 Functors and Natural Transformations

Up until now, we have confined our analysis of category theory merely to that of a system of mappings on objects. We proceed by adopting a more generalized perspective about category theory by considering mappings on categories themselves, and the higher-dimensional constructions that result from such considerations.

Definition 2.13. Let $\mathscr{C}$ and $\mathscr{D}$ be categories. A *functor* F is a mapping

$$F : \mathscr{C} \longrightarrow \mathscr{D}$$

from objects in $\mathscr{C}$ to objects in $\mathscr{D}$ and maps in $\mathscr{C}$ to maps in $\mathscr{D}$ in which for every map $f : X \longrightarrow Y$, the following conditions are satisfied:

1. $F(1_X) = 1_{F(X)}$
2. $F(f : X \longrightarrow Y) = F(f) : F(X) \longrightarrow F(Y)$
3. $F(gf) = F(g)F(f)$

In other words, a functor is a mapping on categories which preserves domains, codomains, identity maps, and the overall composition structure of the category being acted on by the functor.

Example 2.14. In the category **SET**, an example of a functor is that of the power set operation. Given any set X, we can construct its power set $\mathscr{P}(X)$ defined by $\mathscr{P}(X) = \{U \mid U \subseteq X\}$. Then, given any function $f : X \longrightarrow Y$, we can define the function $\mathscr{P}(f) : \mathscr{P}(X) \longrightarrow \mathscr{P}(Y)$ by the direct image operation as follows: For any $U \in \mathscr{P}(X)$, $\mathscr{P}(f)(U) = f[U] = \{f(x) \mid x \in U\}$. This assignment gives rise to the functor $\mathscr{P} : \mathbf{SET} \longrightarrow \mathbf{SET}$. We verify that this is indeed a functor by checking that it preserves identity maps by the following calculation:

$$\begin{aligned}(\mathscr{P}(1_X))(U) &= \{1_X(x) \mid x \in U\} \\ &= \{x \mid x \in U\} \\ &= U \\ &= 1_{\mathscr{P}(X)}(U)\end{aligned}$$

and by checking that it preserves the composition structure by calculating:

$$\begin{aligned}(\mathscr{P}(gf))(U) &= \{gf(x) \mid x \in U\} \\ &= \{g(f(x)) \mid x \in U\} \\ &= \{g(y) \mid y \in \{f(x) \mid x \in U\}\} \\ &= (\mathscr{P}(g))(\{f(x) \mid x \in U\}) \\ &= \mathscr{P}(g)\mathscr{P}(f)(U)\end{aligned}$$

The notion of a functor on categories gives rise to a more generalized species of categories; namely, the category of categories **CAT** whose objects are categories and whose maps are functors between categories.

Definition 2.15. The category of categories **CAT** consists of the following data:

- Objects: given by a set of small categories $\ldots, \mathscr{C}, \mathscr{D}, \mathscr{E}, \ldots$
- Maps: given by a collection of functors on categories $F : \mathscr{C} \longrightarrow \mathscr{D}$
- For every category $\mathscr{C} \in$ **CAT**, there exists an identity functor

$$1_{\mathscr{C}} : \mathscr{C} \longrightarrow \mathscr{C}$$

- For every composable pair of functors $F : \mathscr{C} \longrightarrow \mathscr{D}$, $G : \mathscr{D} \longrightarrow \mathscr{E}$, there exists a composition functor $GF : \mathscr{C} \longrightarrow \mathscr{E}$ defined by

$$(GF)(X) = G(F(X)), \quad (GF)(f) = G(F(f))$$

  for any object X and map $f : X \longrightarrow Y$ in the category $\mathscr{C}$.

It is important to notice that in **CAT**, the objects are given by all and only those categories that are *small* (i.e., those categories in which the collection of objects and collection of maps are both sets, opposed to that of classes). Categories for which the collection of objects and collection of maps are classes are then regarded as large. [1]

With the concept of a functor on categories in mind, which as we just observed, simply turned out to be a structure preserving homomorphism between categories, we arrive at the notion of a natural transformation.

[1]This distinction is due to Russell's observation in his *Principles of Mathematics* (1903), in which he recognized that the set $\{X \mid X \notin X\}$ (i.e., the set of all sets defined by the condition "is not an element of itself,"), is not a set, but rather a class. Assuming that this set, often refereed to as the "Russell set," exists in the universe of sets, implies the existence of a set X such that $X \notin X$ (as we assumed that X is not an element of itself), but it also implies that $X \in X$ as X satisfies the condition of not being an element of itself, hence a contradiction. Therefore, the set of all sets is not a set, but rather a class.

Similarly to that of a functor, a natural transformation is a kind of structure preserving homomorphism, but is one that acts on functors themselves, as opposed to that of categories, so generalizes the notion of a functor on categories which is itself, a generalization of a mapping of objects in a category. Whereas a functor can be rightfully thought of as projecting some sort of picture of its domain into its codomain, natural transformations can be thought of as sliding the picture projected by one functor onto the picture projected by another functor. In particular, we define a natural transformation between functors in the following manner:

Definition 2.16. Let $\mathscr{C}$ and $\mathscr{D}$ be categories and let $F, G : \mathscr{C} \longrightarrow \mathscr{D}$ be a parallel pair of functors. A *natural transformation*

$$\eta : F \longrightarrow G$$

is a family of maps $\eta_X : F(X) \longrightarrow G(X)$ in $\mathscr{D}$, one for each object $X \in \mathscr{C}$ such that for every map $f : X \longrightarrow Y$ in $\mathscr{D}$, the following square commutes

$$\begin{array}{ccc} F(X) & \xrightarrow{\eta_X} & G(X) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(Y) & \xrightarrow[\eta_Y]{} & G(Y) \end{array}$$

Example 2.17. In the category **SET**, an example of a natural transformation is that of the singleton operation $Sing_X : x \longmapsto \{x\}$ with type $X \longrightarrow \mathscr{P}(X)$ which can be defined for any set X. This operation is natural in X in that for any function $f : X \longrightarrow Y$, the following square commutes

$$\begin{array}{ccc} X & \xrightarrow{Sing_X} & \mathscr{P}(X) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \mathscr{P}(f)} \\ Y & \xrightarrow[Sing_Y]{} & \mathscr{P}(Y) \end{array}$$

Note that given any $x \in X$,

$$\begin{aligned}\mathscr{P}(f)(Sing_X)(x) &= (\mathscr{P}(f))(Sing_X(x)) \\ &= (\mathscr{P}(f))(\{x\}) \\ &= \{f\,x\} \\ &= Sing_Y(f\,x) \\ &= (Sing_Y)f(x)\end{aligned}$$

Hence, $Sing_X : X \longrightarrow \mathscr{P}(X)$ is a natural transformation within the parameter X. We may also choose to denote this by $Sing: 1_{\mathbf{SET}} \longrightarrow \mathscr{P}$.

Similarly to the notion of a functor on categories, a natural transformation on functors gives rise to a new species of categories; namely, the category of functors $\mathbf{FUN}(\mathscr{C},\mathscr{D})$ whose objects are functors and whose maps are natural transformations on functors.

Definition 2.18. Given categories $\mathscr{C}$ and $\mathscr{D}$, the functor category $\mathbf{FUN}(\mathscr{C},\mathscr{D})$ consists of the following data:

- Objects: given by functors $F : \mathscr{C} \longrightarrow \mathscr{D}$
- Maps: given by natural transformations $\eta : F \longrightarrow G$ between parallel pairs of functors $F, G : \mathscr{C} \longrightarrow \mathscr{D}$
- Identity: for every functor F, we have the natural transformation

$$(1_F)_X : F(X) \longrightarrow F(X)$$

- Composition: Given any composable pair of natural transformations

$$\eta : F \longrightarrow G, \quad \tau : G \longrightarrow H$$

their composition $\tau\eta : F \longrightarrow H$ is defined by requiring that for $X \in \mathscr{C}$,

$$(\tau\eta)_X = \tau_X \eta_X$$

With the notion of a natural transformation in mind, we are now in a position in which we can define that of a natural isomorphism.

Definition 2.19. A *natural isomorphism* is a natural transformation

$$\eta : F \longrightarrow G$$

which is an isomorphism in the category **FUN**($\mathscr{C}, \mathscr{D}$), i.e.,

$$1_{\mathscr{C}} = GF, \;\; 1_{\mathscr{D}} = FG$$

with $F : \mathscr{C} \longrightarrow \mathscr{D}$ and $G : \mathscr{D} \longrightarrow \mathscr{C}$.

Example 2.20. In the category **SET**, we already know that the power set operation gives rise to a functor. Similarly, the function space

$$2^X = \{f \mid f : X \longrightarrow 2 = \{1, 0\}\}$$

for any set X, also gives rise to a functor; namely that of an exponentiation functor. Then, we can define a natural transformation $\eta_X : \mathscr{P}(X) \longrightarrow 2^X$ which acts as an example of a natural isomorphism in **SET**. In order to see how this works, define $\eta_X(U) = \chi_U$ where χ_U is the characteristic function defined by

$$\chi_U(x) = \begin{cases} 1 & \text{if } \; x \in U \\ 0 & \text{if } \; x \notin U \end{cases}$$

As this is clearly an isomorphism, we can choose its unique inverse function $\eta_X^{-1} : 2^X \longrightarrow \mathscr{P}(X)$ defined by $\eta_X^{-1}(\chi) = \{x \in X \mid \chi(x) = 1\}$. We now wish to verify that η_X is natural in the sense that the following diagram commutes for any function $f : X \longrightarrow Y$

$$\begin{array}{ccc} \mathscr{P}(Y) & \xrightarrow{\eta(Y)} & 2^Y \\ {\scriptstyle \mathscr{P}(f)}\downarrow & & \downarrow{\scriptstyle 2^f} \\ \mathscr{P}(X) & \xrightarrow[\eta(X)]{} & 2^X \end{array}$$

In order to verify the naturality of η_X, recall that for any $V \in \mathscr{P}(X)$, $\mathscr{P}(f)(V) = f^{-1}[V] = \{x \in X \mid f(x) \in V\}$ and that for any function $g \in 2^Y$, $2^f(f) = gf$. Then, for any $V \subseteq Y$ and $x \in X$,

$$\begin{aligned} 2^f \eta_Y(V)(x) &= 2^f(\eta_Y(V))(x) \\ &= 2^f(\chi_V)(x) \\ &= \chi_V f(x) \\ &= \chi_V(f(x)) \\ &= \begin{cases} 1 & \text{if } f(x) \in V \\ 0 & \text{if } f(x) \notin V \end{cases} \end{aligned}$$

Moreover, we calculate

$$\begin{aligned} \eta_X \mathscr{P}(f(V))(x) &= \eta_X(\mathscr{P}(f(V)))(x) \\ &= \eta_X(f^{-1}[V])(x) \\ &= \chi_{f^{-1}[V]}(x) \\ &= \begin{cases} 1 & \text{if } x \in f^{-1}[V] \\ 0 & \text{if } x \notin f^{-1}[V] \end{cases} \end{aligned}$$

Note that these calculations establish the fact that $2^f \eta_Y = \eta_X \mathscr{P}(f)$, thereby establishing the commutativity of the above diagram, so by consequence, the naturality of the isomorphism η_X, as required.

We now arrive at the concept of an equivalence of categories.

Definition 2.21. An *equivalence of categories* between $\mathscr{C}$ and $\mathscr{D}$ consists of a pair of functors

$$F : \mathscr{C} \longrightarrow \mathscr{D}, \;\; G : \mathscr{D} \longrightarrow \mathscr{C}$$

and a pair of natural isomorphisms

$$\eta : 1_{\mathscr{C}} \longrightarrow GF, \;\; \tau : 1_{\mathscr{D}} \longrightarrow FG$$

As η and τ are isomorphisms, we may also choose to simply write $\eta : 1_{\mathscr{C}} \cong GF$ and $\tau : 1_{\mathscr{D}} \cong FG$ respectively. Hence, two categories $\mathscr{C}$ and $\mathscr{D}$ are equivalent (i.e., constitute an equivalence of categories) if there exists an equivalence between them. An equivalence between two categories however needn't be unique. Not only could there exist another pair of functors other than (F,G) which establish this equivalence, there could also exist another pair of natural isomorphisms other than (η,τ). It is important to observe that the notion of an equivalence of categories generalizes the notion of a natural isomorphism of categories in the sense that in the former, we are replacing the identity natural transformations with natural isomorphisms.

Example 2.22. An example of two categories which constitute an equivalence of categories is that of the category **SETFIN** of all finite sets and functions between them and the category **ORDFIN** of finite ordinals and functions between them. We will denote a finite ordinal by $\bar{n} = \{0,1,2,\dots,n-1\}$. To see how this works, let F : **FINORD** $\longrightarrow$ **FINSET** be a functor defined by $F(\bar{n}) = \bar{n}$ and $F(f) = f$. Now define another functor G : **FINSET** $\longrightarrow$ **FINORD** by the following: for every finite set X, set $G(X) = \bar{n}$ in which $n = |X|$. To extend G to mappings, given any function $f : X \longrightarrow Y$ with

$|X| = n$ and $|Y| = m$, we want $G(f) : \bar{n} \longrightarrow \bar{m}$ to behave functorially. To verify this, choose for each finite set X, an arbitrary yet fixed bijection $h_X : X \longrightarrow \bar{n}$ with $n = |X|$. Now define $G(f) : \bar{n} \longrightarrow \bar{m}$ as the unique map which makes the following square commute

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle h_X}\downarrow{\scriptstyle \cong} & & {\scriptstyle \cong}\downarrow{\scriptstyle h_Y} \\ \bar{n} & \xrightarrow[G(f)]{} & \bar{m} \end{array}$$

so satisfies $G(f) = h_Y f h_X^{-1}$. We check that G preserves identity and composition maps by calculating that for any $f : X \longrightarrow Y$ and $g : Y \longrightarrow Z$,

$$\begin{aligned} G(1_X) &= h_X 1_X h_X^{-1} \\ &= h_X h_X^{-1} \\ &= 1_{G(X)} \end{aligned}$$

and

$$\begin{aligned} G(gf) &= h_Z g f h_X^{-1} \\ &= h_Z g h_Y^{-1} h_Y f h_X^{-1} \\ &= G(g)G(f) \end{aligned}$$

We now wish to obtain the required natural isomorphisms. For the natural isomorphism $\eta_{\bar{n}} : GF(\bar{n}) \longrightarrow \bar{n}$, if we set $\eta_{\bar{n}} = h_{\bar{n}}^{-1}$, then the following square commutes

$$\begin{array}{ccc} \bar{n} & \xrightarrow{\eta_{\bar{n}}} & \bar{n} \\ {\scriptstyle G(F(f)) = h_{\bar{m}} f h_{\bar{n}}^{-1}}\downarrow & & \downarrow{\scriptstyle f} \\ \bar{m} & \xrightarrow[\eta_{\bar{m}}]{} & \bar{m} \end{array}$$

and for $\tau_X : FG(X) \longrightarrow X$, if we set $\tau_X = h_X^{-1}$, then the following square commutes

$$\begin{array}{ccc} \bar{n} & \xrightarrow{\tau_X} & X \\ {\scriptstyle F(G(f))=h_Y f h_X^{-1}}\downarrow & & \downarrow{\scriptstyle f} \\ \bar{n} & \xrightarrow[\tau_Y]{} & Y \end{array}$$

Hence, **FINORD** and **FINSET** constitute an equivalence of categories.

Note that in the above example, although **FINSET** and **FINORD** constitute an equivalence of categories, they are not isomorphic. Indeed, every object in **FINORD** is isomorphic to some object in **FINSET**, but the former category has countably many objects whereas the latter category has uncountably many objects. Finally, we arrive at what is perhaps the most fundamental concept within category theory; namely that of an adjunction.

Definition 2.23. Let $\mathscr{C}$ and $\mathscr{D}$ be categories. An *adjunction* consists of a pair of functors $F : \mathscr{C} \longrightarrow \mathscr{D}$, $G : \mathscr{D} \longrightarrow \mathscr{C}$ and a natural transformation

$$\eta : 1_{\mathscr{C}} \longrightarrow GF$$

which satisfies the universal mapping property that for all objects $X \in \mathscr{C}$, $Y \in \mathscr{D}$ and any map $f : X \longrightarrow G(Y)$, there exists a unique map $g : F(X) \longrightarrow Y$ such that $f = G(g)\eta_X$ making the following triangle commute

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & GF(X) \\ & {\scriptstyle f}\searrow & \downarrow{\scriptstyle G(g)} \\ & & G(Y) \end{array}$$

In the above situation, we regard F as being *left adjoint* to G and regard G and being *right adjoint* to F which we denote by $F \dashv G$. Moreover, we regard η as acting as the *unit* of the adjunction. What is being asserted by the above definition is that there exists a functional way in which a map $f : X \longrightarrow G(Y)$ can be converted into a map $g : F(X) \longrightarrow Y$ such that g is a unique solution to the equation $f = G(-)\eta_X$. Hence, an essential property of adjoint functors is that they determine each other uniquely up to isomorphism. That is, if F is a functor which is left adjoint to the functor G which is right adjoint to F, then there exists exactly one natural isomorphism from F to G. It is often useful to represent adjunctions as certain kinds of symmetric inference rules, as in the following

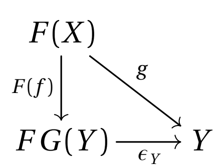
$$\frac{X \longrightarrow G(Y)}{F(X) \longrightarrow Y}$$

The symmetry of definition 2.23 with respect to the functors F and G is captured by the following proposition.

Proposition 2.24. Let $\mathscr{C}, \mathscr{D}$ be categories and let $F : \mathscr{C} \longrightarrow \mathscr{D}$, $G : \mathscr{D} \longrightarrow \mathscr{C}$ be functors such that $F \dashv G$. Then, there exists a natural transformation $\epsilon : FG \longrightarrow 1_{\mathscr{D}}$ (called the *counit* of the adjunction) such that for any map $g : F(X) \longrightarrow Y$, there exists a unique map $f : X \longrightarrow G(Y)$ satisfying

$$g = \epsilon_Y F(f)$$

thereby making the following triangle commute



The above proposition is an immediate consequence of categorical duality, as described in example 2.6 and proposition 2.7. We can dualize the proposition and the resulting adjunction is equivelant to either the natural transformation $\eta : 1_{\mathscr{C}} \longrightarrow GF$ (i.e., the unit of the adjunction) or $\epsilon : FG \longrightarrow 1_{\mathscr{D}}$ (i.e., the counit of the adjunction), along with their respective universal properties. We will delay giving an example of an adjunction until we consider some abstract constructions within categories. As will be seen, a special adjoint will arise out of the notions of a product and exponential object within a category within example 2.36

2.5 Abstract Constructions and the Universal Mapping Property

We proceed by considering some abstract constructions in category theory that will allow us to come to grips with the general structure of a Cartesian closed category. In doing so, we will employ the concept of a universal mapping property, which characterizes certain constructions in a category in terms of their mappings to and from other objects in such a way that the constructions in question are uniquely determined up to isomorphism. The first of these constructions we will consider is that of a terminal object.

Definition 2.25. An object 1 is a *terminal object* in a category $\mathscr{C}$ if and only if it satisfies the universal mapping property that for any object $X \in \mathscr{C}$, there exists a unique map $t_X : X \longrightarrow 1$

Example 2.26. In the category **SET**, the terminal object is given by the singleton set $\{\star\}$, i.e., the set consisting of exactly one element, since it satisfies the universal mapping property that for any set X, there exists a unique function $t_X : X \longrightarrow \{\star\}$.

Example 2.27. In the category **LT-ALG**, the terminal object is given by the truth constant $\top$ as it satisfies the universal mapping property that for any formula ϕ , there exists a proof $\phi \vdash \top$ (i.e., $\phi \leq \top$) which represents the required unique map for terminal objects, by virtue of the inference rule $\top I$.

We no proceed by considering the notion of a binary product in a category.

Definition 2.28. Given objects X, Y in a category $\mathscr{C}$, the object $X \times Y$ is a *product* in $\mathscr{C}$ if and only if it comes equipped with the projection maps

$$\pi_{X,Y} : X \times Y \longrightarrow X, \;\; \pi'_{X,Y} : X \times Y \longrightarrow Y$$

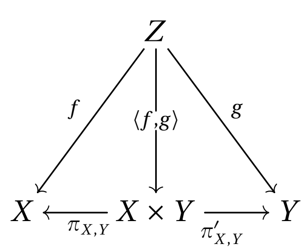
which satisfies the *universal mapping property* that for any object $Z \in \mathscr{C}$ and collection of maps $f : Z \longrightarrow X$, $g : Z \longrightarrow Y$, there exists a unique map

$$\langle f, g \rangle : Z \longrightarrow X \times Y$$

satisfying the following equations

$$\pi_{X,Y} \langle f, g \rangle = f, \;\; \pi'_{X,Y} \langle f, g \rangle = g$$

making the following triangles commute



It is important to observe that this definition does not entail that all categories come equipped with products. Rather, it illustrates what a product looks like in any category in which products in fact exist. If within a category

$\mathscr{C}$, one can define the product of any two objects, we say that $\mathscr{C}$ has *all products* (i.e., all binary products). A concrete example of such a category is that of **SET**.

Proposition 2.29. The category **SET** has all binary products in the sense that a product can be defined between any two sets.

Proof. In the category **SET**, given any two sets X and Y, we can define their product by their Cartesian product by $X \times Y = \{(x, y) \mid x \in X,\ y \in Y\}$ which implies the existence of the (surjective) projections $\pi_{X,Y} : X \times Y \longrightarrow X$ and $\pi'_{X,Y} : X \times Y \longrightarrow Y$ defined by $\pi_{X,Y}(x, y) = x$ and $\pi'_{X,Y}(x, y) = y$ respectively, for every $x \in X,\ y \in Y$. Furthermore, we know that the universal mapping property for products in a category is satisfied by observing that given any set Z and functions $f : Z \longrightarrow X$, $g : Z \longrightarrow Y$, the function $h : Z \longrightarrow X \times Y$ defined by $\langle f(z), g(z)\rangle$ for any $z \in Z$ is the unique function satisfying definition 2.25. We know it is unique since by definition, $\pi_{X,Y}(h(z)) = f(z)$ and $\pi'_{X,Y}(h(z)) = g(z)$, so the function h makes the product diagram in definition 2.25 commute, which implies that $h(z) = \langle \pi_{X,Y} h, \pi'_{X,Y} h\rangle = \langle f(z), g(z)\rangle$. □

Example 2.30. In the category **LT-Alg**, products are given by conjunctions $\phi \wedge \psi$ whose projections are given by the elimination rule for $\wedge$ which induces the existence proofs $\psi \wedge \psi \vdash \psi$ and $\phi \wedge \psi \vdash \psi$. The universal mapping property for products in a category is satisfied since for any formula ω and collection of proofs $\omega \vdash \phi$ and $\omega \vdash \psi$, there exists a proof $\omega \vdash \phi \wedge \psi$ which represents the unique map as required.

More generally, we also have the notion of an arbitrary or finite product of objects (i.e., a product of finitely many objects) in a category. Given a category $\mathscr{C}$ and a finite collection of objects $X_1, \dots, X_n$, we can define their finite product by $X_1 \times \cdots \times X_n$ whose induced projections

$$\pi_i : X_1 \times \cdots \times X_n \longrightarrow X_i$$

map to the i-th factor of the product for $1 \leq i \leq n$. Provided $\mathscr{C}$ comes equipped with a terminal object 1 and has all binary products, one can define products of every finite cardinality, thereby suggesting the following generalization of the notion of a product (i.e., binary product) in a category.

Definition 2.31. A category $\mathscr{C}$ has *all finite products* (or simply, has finite products) if $\mathscr{C}$ comes equipped with a terminal object and has all binary products.

For the purposes of our next definition; namely that of an exponential in a category, it is important to notice that aside from defining the product of two objects in a category, we can also define the product of two maps in a category. In fact, we will soon see that the definition of an exponential in a category depends on not only the existence of products for any two objects, but also the existence of products of maps.

Definition 2.32. Given two maps $f : X \longrightarrow Y$ and $g : X' \longrightarrow Y'$ in a category $\mathscr{C}$, their product $f \times g$ is a map

$$f \times g : X \times X' \longrightarrow Y \times Y'$$

defined by

$$f \times g = \langle f\pi_{X,X}, g\pi'_{X,X} \rangle$$

making the following squares commute

$$\begin{array}{ccccc} X & \xleftarrow{\pi_{X,X}} & X \times X' & \xrightarrow{\pi'_{X,X}} & X' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f \times g} & & \downarrow{\scriptstyle g} \\ Y & \xleftarrow[\pi_{Y,Y}]{} & Y \times Y' & \xrightarrow[\pi'_{Y,Y}]{} & Y' \end{array}$$

Having now made explicit the concept of a product of both objects and maps in a category, we are now in a position in which we can formulate the notion of an exponential of objects in a category.

Definition 2.33. Let $\mathscr{C}$ be a category with all binary products for any two objects X and Y. An object X^Y is an *exponential* in $\mathscr{C}$ if and only if it comes equipped with an evaluation map $\epsilon : X^Y \times Y \longrightarrow X$ which satisfies the universal mapping property that for every object $Z \in \mathscr{C}$ and mapping $f : Z \times Y \longrightarrow X$ in $\mathscr{C}$, there exists a unique map $f^* : Z \longrightarrow X^Y$, which we call the *exponential transpose* of f, satisfying the following equation

$$f = \epsilon(f^* \times 1_Y)$$

thereby making the following triangle commute

$$\begin{array}{ccc} Z \times Y & \xrightarrow{f^* \times 1_Y} & X^Y \times Y \\ & {\scriptstyle f}\searrow & \downarrow{\scriptstyle \epsilon} \\ & & X \end{array}$$

Example 2.34. In the category **SET**, given sets X and Y, their exponential object is given by their function space $X^Y = \{f \in \mathbf{SET} \mid f : Y \longrightarrow X\}$ whose corresponding evaluation function $\epsilon : X^Y \times Y \longrightarrow X$ is defined by $\epsilon(g, y) = g(y)$ for $y \in Y$. Then, for any set Z and function $f : Z \times Y \longrightarrow X$, we define its exponential transpose $f^* : Z \longrightarrow X^Y$ by $f^*(z)(y) = f(z, y)$ and verify that this construction satisfies the universal mapping property by observing that $\epsilon(f^*(z), y) = f(z, y)$.

Example 2.35. In the category **LT-ALG**, given formulas ϕ and ψ, their exponential is given by implications $\phi \Rightarrow \psi$ whose evaluation map is given by the inference rule of $\Rightarrow E$, which we can alternatively represent by the relation $(\phi \Rightarrow \psi) \wedge \phi \vdash \psi$. This satisfies the universal mapping property that for any formula ω and proof $\omega \wedge \phi \vdash \psi$, there exists a proof proof $\omega \vdash \phi \Rightarrow \psi$ which represents the unique map (as required by the previous definition) by virtue of $\Rightarrow I$.

With the concepts of a product and exponential in mind, we return to the notion of an adjunction by giving a concrete example. It will be helpful for the reader to refer back to definition 2.21 before proceeding.

Example 2.36. Let $\mathscr{C}$ be a category with binary products. Then, fix some object $X \in \mathscr{C}$ and consider a product functor $(-) \times X : \mathscr{C} \longrightarrow \mathscr{C}$, defined on objects by $Y \longmapsto Y \times X$ and defined on maps by

$$(g : Y \longrightarrow Z) \longmapsto (g \times 1_X : Y \times X \longrightarrow Z \times X)$$

In order to establish a right adjoint for $(-) \times X$, we would then require the existence of a functor $G : \mathscr{C} \longrightarrow \mathscr{C}$ such that for any $Y, Z \in \mathscr{C}$, there is a natural isomorphism

$$\frac{Y \times X \longrightarrow Z}{Y \longrightarrow G(Z)}$$

In order to achieve this, define G on objects by $G(Z) = Z^X$ and define G on maps by $G(h : Z \longrightarrow W) = h^X : Z^X \longrightarrow W^X$. Setting $G(Z)$ for Y in the above adjoint schema then gives rise to the counit

$$\frac{\epsilon : Z^X \times X \longrightarrow Z}{1_{Z^X} : Z^X \longrightarrow Z^X}$$

which is an adjunction provided that $\epsilon : Z^X \times X \longrightarrow Z$ satisfies the following universal mapping property: given any map $f : Y \times X \longrightarrow Z$, there exists a unique map $f^* : Y \longrightarrow Z^X$ satisfying $f = \epsilon(f^* \times 1_X)$. Note however that this is exactly the universal mapping property for exponentials in a category found in definition 2.33 and so we do in fact have the following adjunction

$$(-) \times X \dashv (-)^X$$

2.6 Cartesian Closed Categories

We now arrive at the second principal concept of this chapter; namely that of a Cartesian closed category.

Definition 2.37. A category $\mathscr{C}$ is *Cartesian closed* if and only if $\mathscr{C}$ comes equipped with a terminal object 1, is closed under binary products $X \times Y$, and exponentials X^Y, for all $X, Y \in \mathscr{C}$.

As will be seen in the proceeding chapters, it is often useful to adopt an equational perspective about categories. We can alternatively define Cartesian closed categories in the following manner.

Definition 2.38. A category $\mathscr{C}$ is Cartesian closed if and only if $\mathscr{C}$ satisfies the following equations:

- For every $f : X \longrightarrow 1$,

$$f = t_X \tag{2.1}$$

- For every $f : Z \longrightarrow X$, $g : Z \longrightarrow Y$, $h : Z \longrightarrow X \times Y$,

$$\pi_{X,Y}\langle f, g\rangle = f \tag{2.2}$$

$$\pi'_{X,Y}\langle f,g\rangle = g \tag{2.3}$$

$$\langle \pi_{X,Y}h, \pi'_{X,Y}h\rangle = h \tag{2.4}$$

- For every $h: Z \times Y \longrightarrow X$, $k: Z \longrightarrow X^Y$,

$$\epsilon\langle h^*\pi_{X,Y}, \pi'_{X,Y}\rangle = h \tag{2.5}$$

$$(\epsilon\langle k\pi_{X,Y}, \pi'_{X,Y}\rangle)^* = k \tag{2.6}$$

The above equations are sufficient in guaranteeing that the above formulation of a Cartesian closed category is in alignment with definition 2.37 in light of the following.

Proposition 2.39. Equations 2.1-2.6 guarantee that within definitions 2.25, 2.28, and 2.33, the respective diagrams commute and the maps which are required to exist are unique.

Proof. Note that equation 2.1 guarantees that any map $f: X \longrightarrow 1$ in $\mathscr{C}$ is equal to the map $t_X: X \longrightarrow 1$ for any any $X \in \mathscr{C}$, which means that t_X is a unique map so guarantees that the object 1 satisfies the universal mapping property for terminal objects in a category. Moreover, equations 2.2 and 2.3 guarantee that the maps $\pi_{X,Y}: X \times Y \longrightarrow X$ and $\pi'_{X,Y}: X \times Y \longrightarrow Y$ are properly constructed projections out of the product $X \times Y$. Then, under the supposition that for any map $h: Z \longrightarrow X \times Y$, we know by equation 2.4 that $h = \langle \pi_{X,Y}h, \pi'_{X,Y}h\rangle = \langle f,g\rangle$ so the pairing $\langle f,g\rangle: Z \longrightarrow X \times Y$ is a unique map in $\mathscr{C}$, guaranteeing that the universal mapping property for products in a category is satisfied, so $X \times Y$ is a product in $\mathscr{C}$. Lastly, equations 2.5 and 2.6 guarantee that any map of the form $h: Z \times Y \longrightarrow X$ induces the existence of a unique exponential transpose $k: Z \longrightarrow X^Y$ which guarantees that the universal mapping property for exponentials in a category is satisfied, so X^Y is an exponential in $\mathscr{C}$. □

In light of our discussion of adjoint functors, it is worth pointing out that Cartesian closed categories can be more succinctly characterized by adjoints; specifically the existence of right adjoints for particular functors.

Proposition 2.40. A category $\mathscr{C}$ is Cartesian closed if and only if the following functors come equipped with right adjoints:

1. $t_{\mathscr{C}} : \mathscr{C} \longrightarrow \mathscr{C}$
2. $\Delta : \mathscr{C} \longrightarrow \mathscr{C} \times \mathscr{C}$
3. $(- \times X) : \mathscr{C} \longrightarrow \mathscr{C}$

for any object $X \in \mathscr{C}$.

Proof. The result is achieved by recognizing that $t_{\mathscr{C}}$ represents the unique functor from the category $\mathscr{C}$ to the terminal category 1, i.e., the category with exactly one object and one map; the identity map. Δ represents the diagonal functor defined on objects by $\Delta(X) = \langle X, X \rangle$ and defined on maps by $\Delta(f) = f \times f$. Note that the right adjoint of $t_{\mathscr{C}}$ is the terminal object and that the right adjoint of Δ is the binary product. Lastly, the right adjoint of $(-) \times X$ is the exponentiation of X. □

Example 2.41. The category **SET** is a Cartesian closed category, which we know by virtue of example 2.26, proposition 2.29, and example 2.34.

Clearly, **LT-ALG** is also another example of a Cartesian closed category for similar reasons. Given the concept of a Cartesian closed category, we proceed by considering functors on Cartesian closed categories which preserve the Cartesian closed structure, known as "Cartesian closed functors".

Definition 2.42. Let $\mathscr{C}, \mathscr{D}$ be Cartesian closed categories. Then, $F : \mathscr{C} \longrightarrow \mathscr{D}$ is a *Cartesian closed functor* if F preserves the Cartesian closed structure. That is, F satisfies the following equations:

- $F(1) = 1$
- $F(X \times Y) = F(X) \times F(Y)$
- $F(X^Y) = F(X)^{F(Y)}$
- $F(t_X) = t_{F(X)}$
- $F(\pi_{X,Y}) = \pi_{F(X),F(Y)}$
- $F(\pi'_{X,Y}) = \pi'_{F(X),F(Y)}$
- $F(\epsilon_{X,Y}) = \epsilon_{F(X),F(Y)}$
- $F(\langle f, g \rangle) = \langle F(f), F(g) \rangle$

With the concept of a Cartesian closed category and a Cartesian closed functor in mind, we end this chapter by considering the category of Cartesian closed categories.

Definition 2.43. The category **CART** of Cartesian closed categories has as objects, Cartesian closed categories and as maps, Cartesian closed functors.

Chapter 3

Model-Theoretical Constructions over Abstract Algebraic Theories

The purpose of this chapter is to describe the connection between category theory and general algebraic theories. As will be seen, categories with finite products are sufficient in offering a categorical semantics for algebraic theories. More importantly, we will investigate the way in which general algebraic theories give rise to a special species of categories known as syntactic categories. This will in effect give us an abstract framework to apply to typed λ-theories in the context of Cartesian closed categories in the proceeding chapter.

3.1 Interpretations of Algebraic Theories

We begin by making explicit the general notion of an algebraic theory.

Definition 3.1. An *algebraic theory* $\mathbb{T}$ is a pair (Σ, A) such that Σ is a *signature*, namely a family of subsets $\{\Sigma_k\}_{k\in\mathbb{N}}$ defined by the following data:

1. The elements of Σ_k consist of k-ary operations
2. The elements of Σ_0 consist of constants (nullary operations)

We then inductively define *terms* over Σ that result in expressions by:

1. Any variable $\ldots, x, y, z, \ldots$ is a term
2. If $(t_1, \ldots, t_k)$ is a k-tuple of terms and $f \in \Sigma_k$ is a k-ary operation, then $f(t_1, \ldots, t_k)$ is a term.

Lastly, A is a set of *axioms* which represents equality (which we take to represent an equivalence relation) between terms constructed over the signature Σ. The terms are said to be *provable* from the axioms of the theory $\mathbb{T}$ in the sense that we have the judgment $\mathbb{T} \vdash t = u$

Example 3.2. A standard example of an algebraic theory is that of a commutative unitial ring, which comes equipped with constants 1 and 0, a unary operation $-$, and two binary operations $+$ and $\cdot$. The equations (axioms) are given by:

1. $(x+y)+z = x+(y+z)$
2. $x+0 = x$
3. $0+x = x$
4. $x+(-x) = 0$
5. $(-x)+x = 0$
6. $x+y = y+x$
7. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
8. $x \cdot 1 = x$
9. $1 \cdot x = x$
10. $(x+y) \cdot z = x \cdot z + y \cdot z$
11. $z \cdot (x+y) = z \cdot x + z \cdot y$
12. $x \cdot y = y \cdot x$

Given the notion of an algebraic theory $\mathbb{T}$, we now want to define the notion of an interpretation of the terms of $\mathbb{T}$, and then eventually that of a model of the terms of $\mathbb{T}$. In order to offer a sufficiently generalized (yet parameterized) definition of such model-theoretical constructions over algebraic theories however, we must in addition, locate ourselves within the abstract algebraic context of a category $\mathscr{C}$ with all finite products (see definition 2.31).

Definition 3.3. Let $\mathbb{T}$ be an algebraic theory and let $\mathscr{C}$ be a category with all finite products. An *interpretation* I of the terms of $\mathbb{T}$ in $\mathscr{C}$ is an object $I(\mathbb{T}) \in \mathscr{C}$ such that for every k-ary operation $f \in \mathbb{T}$, there is a mapping

$$I(f) : I(\mathbb{T})^k \longrightarrow I(\mathbb{T})$$

in $\mathscr{C}$ in which basic constants are mappings of the form

$$I(c) : 1 \longrightarrow I(\mathbb{T})$$

A term t is always interpreted with respect to a *context*, i.e., a collection of variables $x_1, \ldots, x_n$, and we require that the variables in t also appear in the context. We denote that a term t is understood in a context of variables $x_1, \ldots, x_n$, by the judgment $x_1, \ldots, x_n \vdash t$ which is interpreted as a map

$$I(t) : I(\mathbb{T})^n \longrightarrow I(\mathbb{T})$$

which satisfies the following conditions:

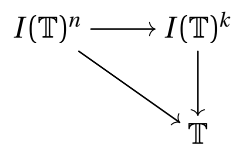
1. The k-th variable x_k is interpreted as the k-th projection map in $\mathscr{C}$

$$\pi_k : I(\mathbb{T})^n \longrightarrow I(\mathbb{T})$$

2. Given a k-tuple of terms $\langle t_1, \ldots, t_k \rangle$, the term $f\langle t_1, \ldots, t_k \rangle$ is interpreted as the following composition map in $\mathscr{C}$

$$I(f)\langle I(t_1), \ldots, I(t_k)\rangle : I(\mathbb{T})^n \longrightarrow \mathbb{T}$$

given explicitly by the following commutative diagram



Note that given the composition map $I(f)\langle I(t_1), \ldots, I(t_k)\rangle$ in the above definition, we can take $I(t_i) : I(\mathbb{T})^n \longrightarrow I(\mathbb{T})$ as being the interpretation of the sub-term t_i for $i \in \mathbb{Z}^+$. Moreover, $I(f)$ is simply the interpretation of the the basic operation f. It therefore follows from the above definition that an interpretation I of a theory $\mathbb{T}$ in a category $\mathscr{C}$ with all finite products is dependent upon the context. For instance, given a context with a single variable x_1, the term $f x_1$ in $\mathbb{T}$ is interpreted by the map $I(f) : I(\mathbb{T}) \longrightarrow \mathbb{T}$ in $\mathscr{C}$ and in a context with two variables x_1, x_2, the term is interpreted as the map $I(f)\pi_1 : I(\mathbb{T})^2 \longrightarrow I(\mathbb{T})$.

It is moreover important to point out that our choice to locate our definition of an interpretation within the context of a category $\mathscr{C}$ with all finite products is motivated by the fact that we require the resulting maps to only act on one argument place. Each context of variables $x_1, \ldots, x_n$ in $\mathbb{T}$ is then taken to be interpreted by an n-fold product in $\mathscr{C}$; namely an object of the form $I(X_1) \times \cdots \times I(X_n)$.

In order to extend the concept of an interpretation I of an algebraic theory $\mathbb{T}$ in a category $\mathscr{C}$ with all finite products to that of a model M, we additionally require the notion of satisfiability within an algebraic theory.

Definition 3.4. Let $x_1, \ldots, x_n$ be a context containing terms t and u. Then, the equality $t = u$ is *satisfied* by the interpretation I provided $I(t) = I(u)$ in the sense that I satisfies the equation $t = u$ if $x_1, \ldots, x_n \vdash t$ and $x_1, \ldots, x_n \vdash u$ are interpreted as the same map in the category $\mathscr{C}$ by I.

Definition 3.5. Let $\mathbb{T}$ be an algebraic theory and let $\mathscr{C}$ be a category with all finite products. A *model* M of $\mathbb{T}$ in $\mathscr{C}$ is an interpretation $I : \mathbb{T} \longrightarrow \mathscr{C}$ that satisfies every axiom in $\mathbb{T}$.

3.2 Models as Functors

Aside from our ability to interpret algebraic theories in the context of a category with all finite products, we can furthermore regard algebraic theories themselves as giving rise to a special species of categories, known as *syntactic* or *equational* categories, built entirely from the algebraic theory itself. The syntactic category then possesses precisely the same algebraic information that comprised the algebraic theory upon which it was constructed. What then follows is that an interpretation of an algebraic theory becomes a functor from the induced syntactic category of the algebraic theory to an arbitrary category with finite products. Hence, we proceed by first considering the way in which an algebraic theory gives rise to its corresponding syntactic category.

Definition 3.6. Given an algebraic theory $\mathbb{T}$, the *syntactic* or *equational category* $S(\mathbb{T})$ constructed from $\mathbb{T}$ has as objects, contexts of variables $[x_1, \ldots, x_n]$ for $n \geq 0$, and has as maps, equivalence classes of n-tuples of terms $\langle t_1, \ldots, t_n \rangle$

which maps a context $[x_1, \ldots, x_m]$ to a context $[x_1, \ldots, x_n]$ which we write as

$$\langle t_1, \ldots, t_n \rangle : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n]$$

and require that each t_k be a term in $\mathbb{T}$ whose variables are included in the context $x_1, \ldots, x_m$. Two maps $\langle t_1, \ldots, t_n \rangle$ and $\langle u_1, \ldots, u_n \rangle$ are equal just in case the axioms of $\mathbb{T}$ prove that $t_k = u_k$ for each $1 \leq k \leq n$. Note that for the sake of convenience, we will not make a notational distinction between terms and their equivalence classes.

Let's now verify that the structure descried in the above definition in fact constitutes a category.

Proposition 3.7. The syntactic category $S(\mathbb{T})$ is in fact a category.

Proof. The identity map $X \longrightarrow X$ is given by n-tuples of the following form

$$\langle x_1, \ldots, x_n \rangle : [x_1 \ldots, x_n] \longrightarrow [x_1, \ldots, x_n]$$

Given any composable pair of maps of the form

$$(t_1, \ldots, t_m) : [x_1, \ldots, x_k] \longrightarrow [x_1, \ldots, x_m]$$

$$(u_1, \ldots, u_n) : [x_1, \ldots, x_m] \longrightarrow [x_1, \ldots, x_n]$$

their composition mapping is given by the n-tuple $(v_1, \ldots, v_n)$ defined by simultaneously substituting the terms $t_1, \ldots, t_m$ for the variables $x_1, \ldots, x_m$ in u_i, as in $v_i = u_i[t_1, \ldots, t_m / x_1, \ldots, x_m]$, for $1 \leq i \leq n$ □

It is important to observe that in addition, we can define the product of contexts $[x_1, \ldots, x_m]$ and $[x_1, \ldots, x_n]$ in $S(\mathbb{T})$ by the context $[x_1, \ldots, x_{m+n}]$, thereby guaranteeing the existence of all finite products in the category

$S(\mathbb{T})$. This is simply a consequence of the fact that any context of variables $[x_1,\dots,x_n]$ in an algebraic theory $\mathbb{T}$ is taken to be an n-fold product $X_1\times\cdots\times X_n$ in $S(\mathbb{T})$. Note here that every object is a product of finitely many copies of $[x_1]$. It should be more clear now that the syntactic category $S(\mathbb{T})$ contains the same algebraic information as it's underlying theory $\mathbb{T}$, as earlier mentioned. This in fact suggests the following alternative formulation of an algebraic theory to that of the one found in definition 3.1.

Definition 3.8. An *algebraic theory* $\mathbb{T}$ is a small category with all finite products in which the objects are a countable sequence of $X^0, X^1, X^2, \dots$ such that $X^m\times X^n = X^{m+n}$ for every $m,n\in\mathbb{N}$. The terminal object 1 is then given by X^0 where every object X is a product of finitely many copies of X^1.

Now that we have established that an algebraic theory $\mathbb{T}$ can just as well be thought of as a syntactic category $S(\mathbb{T})$ with all finite products, we are in a position in which we can refine the notion of a model of an algebraic theory $\mathbb{T}$ in a category $\mathscr{C}$.

Definition 3.9. Let $\mathbb{T}$ be an algebraic theory whose corresponding syntactic category is $S(\mathbb{T})$ and let $\mathscr{C}$ be a category with all finite products. A *model* M of $\mathbb{T}$ in $\mathscr{C}$ is a finite product-preserving functor

$$M : S(\mathbb{T}) \longrightarrow \mathscr{C}$$

defined on objects by:

$$M([x_1,\dots,x_k]) = M(S(\mathbb{T}))^k$$

and defined on mappings by the following conditions:

1. Mappings of the form

$$\langle x_i \rangle : [x_1, \dots, x_k] \longrightarrow [x_1]$$

in $S(\mathbb{T})$ are assigned i-th projections of the form

$$\pi_i : M(S(\mathbb{T}))^k \longrightarrow M(S(\mathbb{T}))$$

in $\mathscr{C}$

2. Mappings of the form

$$\langle f \langle t_1, \dots, t_m \rangle \rangle : [x_1, \dots, x_k] \longrightarrow [x_1]$$

in $S(\mathbb{T})$ are assigned compositions of the form

$$\langle M(f) \langle M(t_1), \dots, M(t_m) \rangle : M(\mathbb{T})^m \longrightarrow \mathbb{T}$$

in $\mathscr{C}$ defined explicitly by the following commutative diagram

$$\begin{array}{ccc} M(\mathbb{T})^m & \longrightarrow & M(\mathbb{T})^k \\ & \searrow & \downarrow \\ & & \mathbb{T} \end{array}$$

such that M assigns to maps of the form

$$\langle t_i \rangle : [x_1, \dots, x_k] \longrightarrow [x_1]$$

in $S(\mathbb{T})$ the value $M(t_i) : M(\mathbb{T})^n \longrightarrow M(\mathbb{T})$ in $\mathscr{C}$ and assigns to every basic operation f in $\mathbb{T}$, the value $M(f)$ in $\mathscr{C}$

3. The mapping

$$\langle t_1, \dots, t_m \rangle : [x_1, \dots, x_k] \longrightarrow [x_1, \dots, x_m]$$

in $\mathbb{T}$ is assigned $\langle M(t_1), \ldots, M(t_m)\rangle$ in $\mathscr{C}$ where M assigns

$$\langle t_i \rangle : [x_1, \ldots, x_k] \longrightarrow [x_1]$$

the value $M(\langle t_i \rangle)$

Observe that the model functor M has been constructed in such a way that it preserves all finite products so can alternatively be thought of as preserving binary products and terminal objects. That $M : S(\mathbb{T}) \longrightarrow \mathscr{C}$ is actually a functor follows from the assumption that M is a model, as this means that all of the equations are satisfied by M. By definition 3.4, the satisfaction of these equations in the syntactic category $S(\mathbb{T})$ implies that they are interpreted by the same map in $\mathscr{C}$.

3.3 Completeness of Algebraic Theories

Now that we have established that a model of an algebraic theory is a functor from the corresponding syntactic category of the algebraic theory being interpreted to an arbitrary category with finite products, we generalize our analysis one step further, by organizing models and a suitable notion of a natural transformation between models into a category of their own right.

Definition 3.10. Let $\mathbb{T}$ be an algebraic theory and let $\mathscr{C}$ be a category with all finite products. The category $\mathbf{MOD}_{\mathscr{C}}(\mathbb{T})$ of *models* of $\mathbb{T}$ in $\mathscr{C}$ is the category whose objects are given by models (functors) of the form $M : S(\mathbb{T}) \longrightarrow \mathscr{C}$ and whose mappings are given by natural transformations $\eta : M \longrightarrow M'$ between models.

Before we arrive at a completeness theorem for algebraic theories, we shall summarize what has been established in this chapter up to this point:

First, we have seen how algebraic theories are just categories, which we called syntactic categories. Such categories were constructed from their underlying theories and contains essentially the same algebraic information as the underlying theory but disregards some of the syntactic information. Hence, we needn't make a notation distinction between an algebraic theory $\mathbb{T}$ and its corresponding syntactic category $S(\mathbb{T})$. Second, we discovered that models are functors from algebraic theories (i.e., syntactic categories) to arbitrary categories with finite products. The functoriality of a model is given to us by virtue of that fact that we require that models satisfy every axiom of the theory. By functoriality however, we also get the preservation of all valid equations in the underlying theory, not just the axioms. Lastly, natural transformations between models is a notion that is immediately guaranteed, since they clearly preserve the entire algebraic structure of the model.

An important fact about models of algebraic theories that have been constructed in this way is that enjoy the property of semantic completeness by virtue of them possessing what is known as a "universal property", meaning that along with satisfying every equation that is provable in the theory, it validates all and only those equations provable from the theory. Hence, we arrive at the following result.

Theorem 3.11. *Given any algebraic theory $\mathbb{T}$, there exists a category $\mathscr{C}$ with all finite products and a model $M \in \boldsymbol{MOD}_{\mathscr{C}}(\mathbb{T})$, which we call the "universal model for $\mathbb{T}$", which satisfies the property that for every equation $t = u$ in $\mathbb{T}$, M satisfies $t = u$ if and only if $\mathbb{T}$ proves $t = u$.*

Proof. The result is achieved by observing the fact that the algebraic theory

$\mathbb{T}$ is a category with all finite products, as in definition 3.8. Hence, if we simply set $\mathbb{T}$ itself as the category $\mathscr{C}$ as in the above statement, models of $\mathbb{T}$ are functors $\mathbb{T} \longrightarrow \mathbb{T}$ that preserve finite products. It then follows that each functor (model) identifies two k-ary operations $f : X^k \longrightarrow X$ and $g : X^k \longrightarrow X$ in $\mathbb{T}$ (as a theory) if and only if $f = g$ (i.e., they represent the same map) in $\mathbb{T}$ as a category. That the terms represent the same map in the category means that they are satisfied by the theory (by definition 3.4). M is thus a universal model in the sense that it satisfies all of the axioms in $\mathbb{T}$ and does not validate anything not provable in $\mathbb{T}$. Note that the "only if" direction gives us completeness and the "if" direction gives us soundness. □

Chapter 4

Categorical Logic

The purpose of this chapter is to make explicit the principal theorem of the thesis; namely the equivalence of categories between the category of Cartesian closed categories and the category of typed λ-theories.

4.1 Cartesian Closed Categories Generated by Typed λ-Calculi

Instead of starting with a general algebraic theory $\mathbb{T}$ as we did in the previous chapter, we now start by considering a particular example of an algebraic theory; namely the algebraic theory Λ of the typed λ-calculus. Recall from the chapter 1 that a typed λ-calculus consists of a set of basic types and a set of basic constants along with their types.

Definition 4.1. An *algebraic theory* Λ *of the typed* λ*-calculus* (or simply, a *typed* λ*-theory*) is a typed λ-calculus along with a set of equations.

Note that in general, any algebraic theory $\mathbb{T}$ gives rise to a typed λ-theory

Λ. There exists a basic type X and for every k-ary operation f, there exists a basic constant $f : X^n \longrightarrow X$ in which X^n is defined by the n-fold product $X_1 \times \cdots \times X_n$ such that $X^0 = 1$, i.e., X^0 is the unit type. We then simply translate the terms of the arbitrary algebraic theory $\mathbb{T}$ to the terms within the associated λ-theory by translating any axiom of $\mathbb{T}$ to its corresponding axiom in the Λ-theory as $x_1 : X_1, \ldots, x_n : X_n \vdash t = u : Y$

We now proceed by applying our notion of an interpretation of a general algebraic theory within the context of a category with all finite products. In order to handle the additional structure that comes with typed λ-theories opposed to that of general algebraic theories, we must in addition require that the category with finite products in which we are working in is also closed under exponentials, so is in particular a Cartesian closed category.

Definition 4.2. Let Λ be a theory of the typed λ-calculus and let $\mathscr{C}$ be a Cartesian closed category. An *interpretation* $[\![-]\!]$ of Λ in $\mathscr{C}$ is given by a mapping $[\![-]\!] : \Lambda \longrightarrow \mathscr{C}$ defined by the following conditions:

1. For every basic type X in Λ, there is assigned an object $[\![X]\!] \in \mathscr{C}$, which is then extended inductively to all types in Λ as follows:

   a) $[\![1]\!] = 1$

   b) $[\![X \times Y]\!] = [\![X]\!] \times [\![Y]\!]$

   c) $[\![X \longrightarrow Y]\!] = [\![Y]\!]^{[\![X]\!]}$

2. For every basic constant c in Λ of type X as in $c : X$, there is assigned a map $[\![c]\!] : 1 \longrightarrow [\![X]\!]$ in $\mathscr{C}$

We then extend the interpretation $[\![-]\!]$ to terms in a context as follows: A context $\Gamma = x_1 : X_1, \ldots, x_n : X_n$ in Λ, is assigned an object $[\![X_1]\!] \times \cdots \times [\![X_n]\!]$ in

$\mathscr{C}$ where if $\Gamma = \emptyset$, then $\llbracket - \rrbracket$ interprets Γ as the terminal object $1 \in \mathscr{C}$. A typing judgment $\Gamma \vdash t : Y$ is then interpreted as a mapping in $\mathscr{C}$

$$\llbracket \Gamma \vdash t : Y \rrbracket : \llbracket X_1 \times \cdots \times X_n \rrbracket \longrightarrow \llbracket Y \rrbracket$$

We then define the interpretation $\llbracket - \rrbracket$ inductively based on a derivation of a valid typing judgment in Λ whose last line is $x_1 : X_1, \ldots, x_n : X_n \vdash t : Y$ by:

1. The interpretation of the k-th variable is the the k-th projection by

$$\llbracket x_0 : X_0, \ldots, x_n : X_n \vdash x_k : X_k \rrbracket = \pi_k : \llbracket \Gamma \rrbracket \longrightarrow \llbracket X_k \rrbracket$$

2. Each basic constant $c : X$ whose context is Γ is then interpreted as the composition map $\llbracket c \rrbracket \llbracket t_\Gamma \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket X \rrbracket$ given explicitly by the diagram

$$\begin{array}{ccc} \llbracket \Gamma \rrbracket & \xrightarrow{\llbracket t_\Gamma \rrbracket} & 1 \\ & \searrow^{\llbracket c \rrbracket \llbracket t_\Gamma \rrbracket} & \downarrow{\scriptstyle \llbracket c \rrbracket} \\ & & \llbracket X \rrbracket \end{array}$$

3. The interpretation of projections and pairing is defined by:

   - $\llbracket \Gamma \vdash \pi_1 t : X \rrbracket = \pi_{X,Y} \llbracket \Gamma \vdash t : X \times Y \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket X \rrbracket$
   - $\llbracket \Gamma \vdash \pi_2 t : Y \rrbracket = \pi'_{X,Y} \llbracket \Gamma \vdash t : X \times Y \rrbracket : \llbracket \Gamma \rrbracket \longrightarrow \llbracket Y \rrbracket$
   - $\llbracket \Gamma \vdash \langle t, u \rangle : X \times Y \rrbracket = \langle \llbracket \Gamma \vdash t : X \rrbracket, \llbracket \Gamma \vdash u : Y \rrbracket \rangle : \llbracket \Gamma \rrbracket \longrightarrow \llbracket X \rrbracket \times \llbracket Y \rrbracket$

4. The interpretation of application and λ-abstraction is defined by:

   - $\llbracket \Gamma \vdash t\, u : Y \rrbracket = \epsilon_{X,Y} \langle \llbracket \Gamma \vdash t : X \longrightarrow Y \rrbracket, [\Gamma \vdash u : X] \rangle : \llbracket \Gamma \rrbracket \longrightarrow \llbracket Y \rrbracket$
   - $\llbracket \Gamma \vdash \lambda x : X . t : X \longrightarrow Y \rrbracket = (\llbracket \Gamma, x : X \vdash t : Y \rrbracket)^* : \llbracket \Gamma \rrbracket \longrightarrow \llbracket Y \rrbracket^{\llbracket X \rrbracket}$

Note that $\epsilon_{X,Y} : \llbracket X \longrightarrow Y \rrbracket \times \llbracket X \rrbracket \longrightarrow \llbracket Y \rrbracket$ is the evaluation of the exponential $\llbracket Y \rrbracket^{\llbracket X \rrbracket}$ and that $(\llbracket \Gamma, x : X \vdash t : Y \rrbracket)^*$ is the exponential transpose of the mapping out of binary products given by $[\![\Gamma, x : X \vdash t : Y]\!] : [\![\Gamma]\!] \times [\![X]\!] \longrightarrow [\![X]\!]$. Although it has already been mentioned in the above definition, it is very important to emphasize the fact that the interpretation $[\![-]\!]$ is inductively based on the derivation of a valid typing judgments in Λ. Moreover, note that every such interpretation of the theory Λ in a Cartesian closed category $\mathscr{C}$ makes every equation defining $\beta\eta$-equivalence true. Similarly to that of the general case in the previous chapter, we move from the notion of an interpretation to that of a model, by requiring the interpretation of the theory Λ satisfies every equation in Λ.

Definition 4.3. An interpretation $[\![-]\!]$ is a *model* of the theory Λ of the typed λ-calculus in a Cartesian closed category $\mathscr{C}$ if and only if $[\![-]\!]$ satisfies all of the axioms in Λ, in the sense that given any axiom $\Gamma \vdash t = u : X$ in Λ, we have $[\![\Gamma \vdash t : X]\!] = [\![\Gamma \vdash u : X]\!]$ in $\mathscr{C}$.

In order to turn a model of the theory Λ into that of a functor (as was done in definition 3.9 the previous chapter), the first step is to verify that the theory Λ gives rise to a corresponding syntactic category.

Definition 4.4. Let Λ be a theory of the typed λ-calculus. Then, the *syntactic category* $S(\Lambda)$ constructed from Λ has objects, the types of Λ as in $1, X, X \times Y, X \longrightarrow Y$, and has as mappings $X \longrightarrow Y$, equivalence classes of terms Λ proves $x : X \vdash t : Y$. Lastly, two maps defined by terms of the form $x : X \vdash t : Y$ and $x : X \vdash u : Y$ in $S(\Lambda)$ are equal (i.e., represent the same map in $S(\Lambda)$) if and only if Λ proves that $x : X \vdash t = u : Y$. The equality relation $=$ is the equivalence relation within definition 1.10.

Let's now verify that $S(\Lambda)$ is in fact a category.

Proposition 4.5. The syntactic category $S(\Lambda)$ constructed from the theory Λ of the typed λ-calculus is a category.

Proof. We first verify that every object X of $S(\Lambda)$ comes equipped with an identity map by observing that Λ proves $x : X \vdash x : X$. Moreover, given any two maps defined by terms of the form $x : X \vdash t : Y$ and $y : Y \vdash u : Z$, their composition map is given by the term $x : X \vdash u[t/y] : Z$ which is the result of substituting t for y in u. Since substitution is unitial and associative, we achieve the desired result. □

Remark 4.6. Note that we take substitution (as given in definition 1.7) in Λ as being interpreted by the composition operation of maps in the corresponding syntactic category $S(\Lambda)$. In particular, if $x : X \vdash t : Y$ and $x : Y \vdash u : Z$ are terms within the theory Λ, then $[\![x : X \vdash t : Y]\!]$ and $[\![x : Y \vdash u : Z]\!]$ are maps in $S(\Lambda)$ which interpret the proved terms respectively, whose composition is given explicitly by

$$[\![x : X \vdash u[t/x] : Z]\!] = [\![x : Y \vdash u : Z]\!][\![x : X \vdash t : Y]\!]$$

In the above situation, the left-hand side of the equality represents the interpretation of $u[t/x]$ in $S(\Lambda)$ and the right-hand side of the equality represents the composition of these interpretations.

In fact, one of the most fundamental results in the categorical semantics for typed λ theories (and algebraic type theories more generally), is that the substitution of a term for a variable in a term can be correctly interpreted as the composition operation of maps in the sense that it preserves types in the required way. This is known as the substitution lemma for categorical semantics and can be formulated in the following manner.

Lemma 4.7. Let $\Gamma' \vdash t : Y$ be a proved term in the theory Λ such that

$$\Gamma' = [x_1 : X_1, \ldots, x_n : X_n]$$

and let $\Gamma \vdash u_i : X_i$ be a proved term for $1 \leq i \leq n$. Then, it follows that $\Gamma \vdash t[u/x] : Y$ is a proved term and moreover, that

$$[\![\Gamma \vdash t[u/x] : Y]\!] = [\![\Gamma' \vdash t : Y]\!]\langle [\![\Gamma \vdash u_1 : X_1]\!], \ldots, [\![\Gamma \vdash u_n : X_n]\!]\rangle$$

Proof. The proof is by induction on the derivation of the typing judgment $\Gamma' \vdash t : Y$. See Crole (1993) and Pitts (1995) for more details. □

Now that we have verified that $S(\Lambda)$ is indeed a category and made explicit how one interprets substitution of a term for a variable in a term as a composition operation of maps, we proceed by verifying that $S(\Lambda)$ is a Cartesian closed category.

Proposition 4.8. The syntactic category $S(\Lambda)$ is a Cartesian closed category.

Proof. We proceed by verifying that $S(\Lambda)$ satisfies our equational representation of a Cartesian closed category found in definition 2.38. First, observe that the terminal object is given by the unit type 1, since for any type X, we have

$$x : X \vdash \star : 1$$

which we know is unique (as required by the universal mapping property for terminal objects in a category) because we have

$$\Gamma \vdash t = \star : 1$$

as an axiom in the theory Λ for any term of unit type 1. Moreover, given types X and Y, their product is given by their product type $X \times Y$ whose corresponding projection maps are given by equivalence classes of terms of the form

$$x : X \times Y \vdash \pi_1 x : X, \;\; x : X \times Y \vdash \pi_2 x : Y$$

We know that product type satisfies the universal mapping property for products in a category since for any type X and collection of terms

$$z : Z \vdash t : X, \;\; z : Z \vdash t : Y$$

there is a unique term $z : Z \vdash \langle t, u \rangle : X \times Y$ (representing the unique map) satisfying

$$z : Z \vdash \pi_1 \langle t, u \rangle = t : X, \;\; z : Z \vdash \pi_2 \langle t, u \rangle = u : Y$$

where

$$s = \langle \pi_1 s, \pi_2 s \rangle = \langle t, u \rangle$$

if $\pi_1 s = t$ and $\pi_2 s = u$. Lastly, given types X and Y, their exponential is given by their function type $X \longrightarrow Y$ whose evaluation mapping is given by

$$v : (X \longrightarrow Y) \times X \vdash (\pi_1 v)(\pi_2 v) : Y$$

We know that the universal mapping property for exponentials in a category is satisfied since given any term $v : Z \times X \vdash t : Y$, the exponential transpose is the term

$$z : Z \vdash \lambda x : X.(t[\langle z, x \rangle / v]) : X \longrightarrow Y$$

which we know is in fact the exponential transpose of t by the following calculation

$$\begin{aligned}
(\lambda x : X.(t[\langle \pi_1 v, x \rangle / v]))(\pi_2 v) &= t[\langle \pi_1 v, \pi_2 v \rangle / v] \\
&= t[v/v] \\
&= t
\end{aligned}$$

We know that the exponential transpose is unique since given any term of the form $z : X \vdash s : X \longrightarrow Y$ such that

$$(s[\pi v/z])(\pi' v) = t$$

is equal to $\lambda x : X.(t[\langle z, x\rangle/v])$, which we verify by calculating

$$\begin{aligned} t[\langle z, x\rangle/v] &= (s[\pi v/z])(\pi' v)[\langle z, x\rangle/v] \\ &= (s[\pi\langle z, x\rangle/z])(\pi\langle z, x\rangle) \\ &= (s[z/z])x \\ &= s\,x \end{aligned}$$

which implies that

$$\begin{aligned} \lambda x : X.(t[\langle z, x\rangle/v]) &= \lambda x : X.(s\,x) \\ &= s \end{aligned}$$

Since the category $S(\Lambda)$ is closed under products, exponentials, and comes equipped with a terminal object, $S(\Lambda)$ is a Cartesian closed category. □

Recall that a Cartesian closed functor is a functor that in addition to preserving the objects, maps, and composition structure of the category in question, preserves the Cartesian closed structure of the category. Please refer back to the definition of a Cartesian closed functor found in chapter 2. Hence, establishing the syntactic category $S(\Lambda)$ then allows us to define models as Cartesian closed functors just as we did in the previous chapter (minus the requirement that the model preserve the exponentials of course).

Definition 4.9. A *model* $[\![-]\!]$ of an algebraic theory Λ of the typed typed λ-calculus in a Cartesian closed category $\mathscr{C}$ is a Cartesian closed functor

$$[\![-]\!] : S(\Lambda) \longrightarrow \mathscr{C}$$

so preserves finite products and exponentials (i.e., binary products, terminal objects, and exponentials).

This allows us to then proceed to a completeness theorem for for Cartesian closed categories with respect to typed λ-theories. Again, this result is demonstrated in same way in which we proved completeness for the case of categories with all finite products with respect to general algebraic theories. That is, there exists a canonical interpretation of Λ in its corresponding syntactic category $S(\Lambda)$, which contains precisely the same algebraic information as the theory Λ upon which it was constructed. This canonical interpretation interprets every basic type X as itself and every basic constant $c : X$ as $x : 1 \vdash x : X$. We can then simply refer to the canonical interpretation as a *syntactic model*, which we know to be a universal model for Λ by the following:

Theorem 4.10. *Let $S(\Lambda)$ be the syntactic category constructed from Λ. Then, for any terms $\Gamma \vdash t : X$ and $\Gamma \vdash u : X$, Λ proves $\Gamma \vdash t = u : X \iff$ the syntactic category $S(\Lambda)$ satisfies $\Gamma \vdash t = u : X$, i.e., if and only if t and u represent the same map in $\mathscr{E}(\Lambda)$.*

Proof. This result is demonstrated in precisely the same way as it was in the case for general algebraic theories (see theorem 3.11). Any λ-theory has a canonical interpretation in its corresponding syntactic category $\mathscr{E}(\Lambda)$ in which every basic type X is interpreted as as itself and interprets every basic constant c with type X as the map $x : 1 \vdash c : X$. The canonical interpretation acts as a model of Λ and is in fact a universal model, since by virtue of the way in which the syntactic category $S(\Lambda)$ was constructed, it follows that for any terms $\Gamma \vdash : X$ and $\Gamma \vdash u : X$, Λ proves $\Gamma \vdash t = u : X \iff$ the syntactic category $S(\Lambda)$ satisfies $\Gamma \vdash t = u : X$, i.e., if and only if t and u represent the same map in $S(\Lambda)$. $\square$

4.2 Typed λ-Calculi Generated by Cartesian Closed Categories

We now proceed by generalizing algebraic theories Λ of the typed λ-calculus into a category of their own, by defining a suitable notion of a mapping between such theories.

Definition 4.11. Let Λ and Λ' be typed λ-theories as defined in 4.1. A *translation homomorphism* is a map $\theta : \Lambda \longrightarrow \Lambda'$ defined by the following conditions:

1. For every basic type $X \in \Lambda$, there is assigned a type $\theta(X) \in \Lambda'$. We then extend θ to all types in Λ by the following equations:

   a) $\theta(1) = 1$

   b) $\theta(X \times Y) = \theta(X) \times \theta(Y)$

   c) $\theta(X \longrightarrow Y) = \theta(X) \longrightarrow \theta(Y)$

2. For every basic constant c in Λ with type X, there is assigned a term $\theta(c)$ with type $\theta(X)$ in Λ'. We then extend θ to all terms in Λ by the following equations:

   a) $\theta(\pi_1 t) = \pi_1(\theta(t))$

   b) $\theta(\pi_2 t) = \pi_2(\theta(t))$

   c) $\theta\langle t, u\rangle = \langle \theta(t), \theta(u)\rangle$

   d) $\theta(t\,u) = \theta(t)\theta(u)$

   e) $\theta(\lambda x : X.t) = \lambda x : \theta(X).\theta(t)$

   f) $\theta(x) = x$ if $x \in Var$

3. For every context $\Gamma = x_1 : X_1, \ldots, x_n : X_n$ in Λ, there is assigned in Λ'

$$\theta(\Gamma) = x_1 : \theta(X_1), \ldots, x_n : \theta(X_n)$$

4. Lastly, we require that θ preserve the axioms of Λ in the sense that if

$$\Gamma \vdash t = u : X$$

   is an axiom in in Λ, then then there exists a proof of

$$\theta(\Gamma) \vdash \theta(t) = \theta(u) : \theta(X)$$

   in Λ', thereby guaranteeing that θ translates all equations provable within Λ to equations that are provable in Λ'.

Given the notion of a translation homomorphism between typed λ-theories, we can arrange such theories into a category of their own right.

Definition 4.12. The category $\mathbf{CALC}_\Lambda$ has as objects, typed λ-theories Λ and as mappings, translation homomorphisms $\theta : \Lambda \longrightarrow \Lambda'$.

Clearly, the identity maps in $\mathbf{CALC}_\Lambda$ are given by translation homomorphisms of the form $1_\Lambda : \Lambda \longrightarrow \Lambda$ and given any composable pair of translations $\theta : \Lambda \longrightarrow \Lambda'$ and $\theta' : \Lambda' \longrightarrow \Lambda''$, their composition is given by the translation of the form $\theta'\theta : \Lambda \longrightarrow \Lambda''$. We now proceed in the reverse direction relative to section 4.1 by constructing typed λ-theories (with arbitrarily large languages) from Cartesian closed categories, resulting in what is known as an *internal language* of a Cartesian closed category.

Definition 4.13. Let $\mathscr{C}$ be a small Cartesian closed category. Then, there exists a theory of the typed λ-calculus $\mathbb{L}(\mathscr{C})$ called the *internal language* of $\mathscr{C}$ defined by the following conditions:

1. For every object $X \in \mathscr{C}$, there is assigned a basic type $\ulcorner X \urcorner \in \mathbb{L}(\mathscr{C})$
2. For every map $f : X \longrightarrow Y$ in $\mathscr{C}$, there is assigned a basic constant $\ulcorner f \urcorner$ with type $\ulcorner X \urcorner \longrightarrow \ulcorner Y \urcorner$ in $\mathbb{L}(\mathscr{C})$
3. For every object $X \in \mathscr{C}$, there is assigned an axiom
$$x : \ulcorner X \urcorner \vdash \ulcorner 1_X \urcorner x = x : \ulcorner X \urcorner$$
4. For all maps $f : X \longrightarrow Y$, $g : Y \longrightarrow X$, and $h : X \longrightarrow Z$ in $\mathscr{C}$ such that $h = gf$, there is assigned an axiom
$$x : \ulcorner X \urcorner \vdash \ulcorner h \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner Z \urcorner$$
5. There exists a constant $T : 1 \longrightarrow [\![1]\!]$ and moreover, for every $X, Y \in \mathscr{C}$, there exists constants
$$\Pi_{X,Y} : \ulcorner X \urcorner \times \ulcorner Y \urcorner \longrightarrow \ulcorner X \times Y \urcorner, \quad E_{X,Y} : (\ulcorner X \urcorner \longrightarrow \ulcorner Y \urcorner) \longrightarrow \ulcorner Y^X \urcorner$$
that satisfy the following conditions which guarantees the desired Cartesian closed structure:
   - $t : \ulcorner 1 \urcorner \vdash T\star = t : \ulcorner 1 \urcorner$
   - $x : \ulcorner X \times Y \urcorner \vdash \Pi_{X,Y} \langle \ulcorner \pi_0 \urcorner z, \ulcorner \pi_1 \urcorner z \rangle = z : \ulcorner X \times Y \urcorner$
   - $w : \ulcorner X \urcorner \times \ulcorner Y \urcorner \vdash \langle \ulcorner \pi_0 \urcorner (\Pi_{X,Y} w), \ulcorner \pi_1 \urcorner (\Pi_{X,Y} w) \rangle = w : \ulcorner X \urcorner \times \ulcorner Y \urcorner$
   - $f : \ulcorner Y^X \urcorner \vdash E_{X,Y}(\lambda x : \ulcorner X \urcorner . (\ulcorner \epsilon_{X,Y} \urcorner (\Pi_{X,Y} \langle f, x \rangle))) = f : \ulcorner Y^X \urcorner$

- $f : \ulcorner X \urcorner \to \ulcorner Y \urcorner \vdash \lambda x : \ulcorner X \urcorner.(\ulcorner \epsilon \urcorner(\Pi_{X,Y}\langle(E_{X,Y} f), x\rangle)) = f : \ulcorner X \urcorner \to \ulcorner Y \urcorner$

Remark 4.14. We make use of the constants T, Π and E for terminals, projection, and exponentiation respectively to ensure the desired isomorphisms are satisfied:

$$[\![1]\!] \cong 1, \;\; [\![X \times Y]\!] \cong [\![X]\!] \times [\![Y]\!], \;\; [\![Y^X]\!] \cong [\![X]\!] \to [\![Y]\!]$$

For instance, the constant $\Pi_{X,Y}$ does not map product types directly to products in the category. Rather, it maps products to basic types of a product which then in turn corresponds to products in the category. likewise, $E_{X,Y}$ maps function types to to basic types of a function type which then corresponds to the exponential in the category, and similarly for unit types. Note that we define isomorphisms between types in the following manner.

Definition 4.15. Let X and Y be types and let Λ-theory. Then X are Y are isomorphic if there exist terms $x : X \vdash t : Y$ and $y : Y \vdash u : X$ such that $x : X \vdash u[t/y] = x : X$ and $y : Y \vdash t[u/x] = y : Y$ are provable from $\mathbb{T}$.

In summary what we have is that of a functor $\mathbb{L}$:**CART**$\longrightarrow$**CALC**$_\Lambda$ where if we recall from definition 2.43, **CART** is the category of small Cartesian closed categories whose objects are given by Cartesian closed categories and whose mappings are given by Cartesin closed functors which preserve binary products, exponentials, and terminal objects. This is due to the fact that given any Cartesian closed functor $F : \mathscr{C} \longrightarrow \mathscr{D}$, then $\mathbb{L}(F) : \mathbb{L}(\mathscr{C}) \longrightarrow \mathbb{L}(\mathscr{D})$ is a translation homomorphism which assigns $\ulcorner F(X) \urcorner$ to each basic type $\ulcorner X \urcorner$, assigns $\ulcorner F(f) \urcorner$ to each basic constant $\ulcorner f \urcorner$, and assigns $\Pi_{F(X),F(Y)}$, $E_{F(X),F(Y)}$, T to the constants $\Pi_{X,Y}$, $E_{X,Y}$, and T respectively.

Similarily, in the other direction, we observed how from a typed λ-theory Λ, we can form the syntactic category $\mathscr{E}(\Lambda)$. This then gives rise to the functor S :$\mathbf{CALC}_\Lambda \longrightarrow \mathbf{CART}$ because given any translation $\theta : \Lambda \longrightarrow \Lambda$, there is induced a functor $S(\theta) : S(\Lambda) \longrightarrow S(\Lambda)$ which sends basic types $X \in S(\Lambda)$ to objects $\theta(X) \in S(\Lambda)$, sends basic constants of the form $x : 1 \vdash c : X$ to maps of the form $x : 1 \vdash \theta(c) : X$, and is then defined on $S(\Lambda)$ by induction on the types and terms.

4.3 An Equivalence of Categories

We are now in a position in which we can establish the principal theorem of this work. We already known what it means to have an equivalence of categories. However, our result also relies on the weaker notion of an equivalence of general algebraic theories which we characterize in the following manner.

Definition 4.16. An *equivelance* of algebraic theories $\mathbb{T}$ and $\mathbb{T}'$ is given by a pair of translations $F : \mathbb{T} \longrightarrow \mathbb{T}'$, $G : \mathbb{T}' \longrightarrow \mathbb{T}$ such that for any $X \in \mathbb{T}$ and $Y \in \mathbb{T}'$,

$$G(F(X)) \cong X, \quad F(G(Y)) \cong Y$$

With all this in mind, we arrive at the principal theorem of this thesis.

Theorem 4.17. *The functors* $\mathbb{L} : \mathbf{CART} \longrightarrow \mathbf{CALC}_\lambda$ *and* $\mathscr{E} : \mathbf{CALC}_\lambda \longrightarrow \mathbf{CART}$ *give rise to an equivelance of categories in the sense that for any Cartesian closed category* $\mathscr{C} \in \mathbf{CART}$, *there is an equivalence of categories*

$$\mathscr{C} \cong \mathscr{E}(\mathbb{L}(\mathscr{C}))$$

and for any typed λ-theory $\Lambda \in \mathbf{CALC}_\lambda$, there is an equivalence of theories

$$\Lambda \cong \mathbb{L}(S(\Lambda))$$

Proof. Let $\mathscr{C}$ be a small Cartesian closed category and let $\eta : \mathscr{C} \longrightarrow \mathscr{E}(\mathbb{L}(\mathscr{C}))$ be a functor defined by the condition that for every object $X \in \mathscr{C}$ and map $f : X \longrightarrow Y$ in $\mathscr{C}$,

$$\eta(X) = \ulcorner X \urcorner, \;\; \eta(f) = (X : \ulcorner X \urcorner \vdash \ulcorner f \urcorner x : \ulcorner Y \urcorner)$$

Since $\mathbb{L}(\mathscr{C})$ proves that for every object X and collection of maps $f : X \longrightarrow Y$, $g : Y \longrightarrow Z$ in $\mathscr{C}$,

$$x : \ulcorner X \urcorner \vdash \ulcorner 1_X \urcorner x = x : \ulcorner X \urcorner$$

and

$$x : \ulcorner X \urcorner \vdash \ulcorner g f \urcorner x = \ulcorner g \urcorner (\ulcorner f \urcorner x) : \ulcorner Z \urcorner$$

we know that η is a properly constructed functor. In order to achieve the desired equivalence in regards to the functor $\eta : \mathscr{C} \longrightarrow S(\mathbb{L}(\mathscr{C}))$, it is sufficient to show that for every object $X \in S(\mathbb{L}(\mathscr{C}))$, there exists an object $\alpha(X) \in \mathscr{C}$ such that $\eta(\alpha(X)) \cong X$, where α is a choice function defined by the following inductive clauses:

1. $\alpha(1) = 1$
2. $\alpha(\ulcorner X \urcorner) = X$
3. $\alpha(X \times Y) = \alpha(X) \times \alpha(Y)$
4. $\alpha(X \longrightarrow Y) = \alpha(Y)^{\alpha(X)}$

We know that this is in alignment with our understanding of an equivalence of categories, since we can extend α to that of a functor $\alpha : S(\mathbb{L}(\mathscr{C})) \longrightarrow \mathscr{C}$ such that

$$\alpha\eta \cong 1_{\mathscr{C}}, \ \ \eta\alpha \cong 1_{S(\mathbb{L}(\mathscr{C}))}$$

Then, given the theory Λ of the typed λ-calculus, define a translation functor

$$\theta : \Lambda \longrightarrow \mathbb{L}(S(\Lambda))$$

by the following conditions:

1. For every basic type $X \in \Lambda$,

$$\theta(X) = \ulcorner X \urcorner$$

2. For every basic constant $c : X$,

$$\theta(\ulcorner x : 1 \vdash c : \theta(X) \urcorner)$$

For the inverse direction, we define a translation

$$\beta : \mathbb{L}(S(\Lambda)) \longrightarrow \Lambda$$

by the following conditions:

1. For every basic type $\ulcorner X \urcorner \in \mathbb{L}(S(\Lambda))$, $\beta(\ulcorner X \urcorner) = X$

2. Given a type $\ulcorner X \to Y \urcorner$ with basic constant $\ulcorner x : X \vdash t : Y \urcorner$,

$$\beta(\ulcorner x : X \vdash t : Y \urcorner) = \lambda x : X.t$$

3. β then acts on the basic constants T, $\Pi_{X,Y}$, $E_{X,Y}$, as follows:

   a) $\beta(T) = \lambda x : 1.x$

b) $\beta(\Pi_{X,Y}) = \lambda p : X \times Y.p$

c) $\beta(E_{X,Y}) = \lambda f : X \longrightarrow Y.f$

What results in the construction immediately above, is that for any type $x \in \Lambda$, $\beta(\theta(X)) \cong X$. For the original direction involving , we then want to similarily show that for any type $X \in \mathbb{L}(S(\Lambda))$, $\theta(\beta(X)) \cong X$. This is demonstrated through a structural induction on the complexity of types:

1. If $A = 1$, then $\theta(\beta(1)) = 1$

2. If $A = \ulcorner X \urcorner$ is a basic type, then $X \in \Lambda$. We now proceed to the inductive step on the complexity of X.

   a) If $X = 1$, then

   $$\theta(\beta(\ulcorner 1 \urcorner)) = 1$$

   We know that there exists an isomorphism $1 \cong \ulcorner 1 \urcorner$ by virtue of the constant $T : 1 \longrightarrow \ulcorner 1 \urcorner$

   b) If X is a basic type, then

   $$\theta(\beta(\ulcorner X \urcorner)) = \ulcorner X \urcorner$$

   c) If $X = Y \times Z$, then

   $$\theta(\beta(\ulcorner Y \times Z \urcorner)) = \ulcorner Y \urcorner \times \ulcorner Z \urcorner$$

   We know that there exists an isomorphism $\ulcorner Y \times Z \urcorner \cong \ulcorner Y \urcorner \times \ulcorner Z \urcorner$ by the constant

   $$\Pi_{X,Y} : \ulcorner Y \urcorner \times \ulcorner Z \urcorner \longrightarrow \ulcorner Y \times Z \urcorner$$

d) If $X = Y \to Z$, then

$$\theta(\beta(\ulcorner Y \to Z \urcorner)) = \ulcorner Y \urcorner \to \ulcorner Z \urcorner$$

we know that there exists an isomorphism $\ulcorner Y \to Z \urcorner \cong \ulcorner Y \urcorner \to \ulcorner Z \urcorner$ by the constant

$$E_{X,Y} : \ulcorner Y \urcorner \to \ulcorner Z \urcorner = \ulcorner Y \to Z \urcorner$$

3. If $X = Y \times Z$, then

$$\theta(\beta(Y \times Z)) = \theta(\beta(Y)) \times \theta(\beta(Z))$$

Then, by our induction hypothesis, we know that

$$\theta(\beta(Y)) \cong Y, \quad \theta(\beta(Z)) \cong Z$$

which implies that

$$\theta(\beta(Y)) \times \theta(\beta(Z)) \cong Y \times Z$$

4. If $X = Y \to Z$, then

$$\theta(\beta(X \to Y)) = \theta(\beta(Y)) \to \theta(\beta(Z))$$

Then by our induction hypothesis, we know that

$$\theta(\beta(Y)) \cong Y, \quad \theta(\beta(Z)) \cong Z$$

which implies that

$$\theta(\beta(Y)) \to \theta(\beta(Z)) \cong Y \to Z$$

□

4.4 Further Directions

Having examined the equivalence that can be demonstrated between the category of Cartesian closed categories and the category of typed λ-theories, we conclude by mentioning some related concepts which enrich this categorical extension of the Curry-Howard isomorphism. First, note that when we considered the Curry-Howard isomorphism, we limited ourselves to the positive fragment of the intuitionistic propositional calculus, including the truth constant, conjunction, and implication. The Curry-Howard isomorphism can in fact be extended to to the full intuitionistic propositional calculus including disjunction and the false constant, which corresponds to sum type and empty type respectively within the typed Λ-calculus.

Similarly, in regards to the categorical aspects of this work, we limited ourselves to the study of Cartesian closed categories. However, these structures can be extended to that of bicartesian closed categories which in addition to the standard Cartesian closed structure, come equipped with coproducts and and an initial object. These end up corresponding to the sum types and empty type within the typed λ-calculus. This categorical extension of the Curry-Howard isomorphism can therefore be extended to that of the full intuitionistic propositional calculus, the typed λ-calculus including sum type and an empty type, along with bicartesian closed categories. It is even possible to round off this correspondence between the intuitionistic propositional calculus, the typed λ-calculus, and bicartesian closed categories, by directly establishing a correspondence between the intuitionistic propositional calculus and bicartesian closed categories. Here, truth constants are taken to correspond to terminal objects, the false constant is taken to correspond to initial objects, conjunctions correspond to products, disjunctions correspond to coproducts, and implications correspond to exponential objects.

4.5 Conclusion

In this work, we have investigated how Cartesian closed categories extend the Curry-Howard isomorphism. The technique by which this correspondence was demonstrated was seen to be heavily reliant on our ability to interpret λ-theories as certain kinds of algebraic theories, which allowed us to treat Cartesian closed categories and typed λ-theories as interdefinable structures. This is but one example of how the language of category theory is in the unique position of being able to associate various species of mathematical structures which are at face value, drastically distinct.

Bibliography

Awodey, Steve. 2010. *Category Theory.* New York: Oxford University Press.

Barr, Michael and Charles Wells. 1995. *Category Theory for Computing Science.* Upper Saddle River: Prentice Hall.

Brouwer, L.E.J. 1907. On the Foundations of Mathematics. Ph.D. thesis, University of Amsterdam.

Church, Alonzo. 1932. A Set of Postulates for the Foundations of Logic. *Annals of Mathematics* 33(2): 346–366.

Crole, Roy. 1993. *Categories for Types.* Cambridge: Cambridge University Press.

Curry, Haskell. 1934. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences in the United States* 20(11): 584–590.

Curry, Haskell and Robert Feys. 1958. *Combinatory Logic.* Amsterdam: North-Holland Publishing.

Eilenberg, Samuel and Saunders MacLane. 1945. General theory of natural equivalences. *Transactions of the American Mathematical Society* 58(2): 231–294.

Gentzen, Gerhard. 1935a. Untersuchungen über das logische Schlieben I. *Mathematische Zeitschrift* 39: 176–210. English Translation in Szabo (1969), pp. 68-131.

Gentzen, Gerhard. 1935b. Untersuchungen über das logische Schlieben II. *Mathematische Zeitschrift* 405–431. English Translation in Szabo (1969), pp. 68-131.

Heyting, Arend. 1930. Die formalen regeln der intuitionistichen logik. *unknown* 1–150.

Howard, William. 1969. The formulae-as-types notion of construction. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, eds. Jonathan Seldin and Roger Hindley, 479–480. Boston: Academic Press.

Lambek, Joachim. 1968. Deductive Systems and Categories I. *The Journal of Mathematical Systems Theory* 2: 278–318.

Lambek, Joachim. 1969. Deductive Systems and Categories II. *Springer LNM* 274: 76–122.

Lambek, Joachim. 1972. Deductive Systems and Categories III. *Springer LNM* 86: 57–82.

Lambek, Joachim. 1974. Functional Completeness and Cartesian Categories. *Annals of Mathematical Logic* 6: 259–292.

Lambek, Joachim and Phillip Scott. 1986. Introduction to Higher Order Categorical Logic. *Cambridge Studies in Advanced Mathematics* 7: 1–293.

Lawvere, William. 1963. Functorial sematics of algebraic theories. *Proceedings of the National Academy of Sciences U.S.A.* 50: 869–972.

Lawvere, William. 1969a. Adjointness in foundations. *Dialectica* 23: 281–296.

Lawvere, William. 1969b. Diagonal argument in cartesian closed categories. *Category Theory, Homology Theory and their Applications II (Springer LNM)* 92: 134–145.

MacLane, Saunders. 1971. *Categories for the Working Mathematician.* New York: Springer–Verlag.

Pitts, Andrew. 1995. Categorical logic. In *Handbook of Logic in Computer Science: Volume 6*, ed. S. Abramsky. Oxford: Oxford University Press.

Russell, Bertrand. 1903. *The Principles of Mathematics.* Cambridge: Cambridge University Press.

Sorensen, Morten and Pawel Urzyczyn. 2006. *Lectures on the Curry-Howard Isomorphism in Studies in Logic and the Foundations of Mathematics, vol. 149.* Amsterdam: Elsevier.

Van Atten, Mark. 2017. The Development of Intuitionistic Logic. URL <https://plato.stanford.edu/archives/win2017/entries/intuitionistic-logic-development/>.