Cartesian closed categories and the simply-typed λ-calculus

Third year project

Nick Hu

St. Catherine's College

University of Oxford

Supervised by Professor Luke Ong

A project submitted for the degree of

MCompSci Computer Science

Trinity 2018

Abstract

The simply-typed λ-calculus provides a basis for modern-day typed functional programming languages like Haskell; however, it is not immediately clear what the mathematical meaning of such a structure is. Lambek and Scott showed that one way to naturally capture the meaning is with the notion of Cartesian closed category. We analyse one such approach, first developing an algebraic theory of simply-typed λ-calculus, and then attributing to it a categorical model. Then we develop a soundness theorem, to substantiate the claim that such a model is valid. Finally, we develop a completeness theorem: for every theory there exists a 'minimal' unique category which models it. We discuss at length the proposition that the simply-typed λ-calculus is *the same* as Cartesian closed categories.

Contents

1 INTRODUCTION

In the most general sense, a computer program is a structured collection of instructions which yield some result by computation. Here is an example:

Listing 1. A basic factorial functional program.

```
factorial : nat → nat
factorial 0 = 1
factorial n = n * factorial (n-1)
```

We might describe the program itself as a *syntactic object* (a string of text and symbols), which contains *computational content* (a procedure to compute the factorial function). Intuitively, as programmers, we know that this is notionally the same as the factorial function on natural numbers; that is, a mapping

$$\begin{array}{rccc} ! & : & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & & n & \longmapsto & n \times (n-1) \times \cdots \times 1 \end{array}$$

which we describe as a *denotation* — a mathematical object associated to the program from before. In this project, we will explore certain denotations of certain programs.

1.1 The meaning of computer programs

Formal semantics of programming languages concerns itself with using formal mathematics to interpret the *meaning* of programming languages: broadly speaking, it gives us a *semantics* to understand the *syntax* of a given language. There are three main methodologies in this field:

Axiomatic semantics starts with observing the syntax, and attributing properties to program fragments, defined in some program logic. Assertions (invariants) about such properties which hold under every execution form the base of the semantics, and proofs of correctness utilise only those assertions to give an argument independent of any underlying implementation details. Two different implementations of the same algorithm — which are subject to the same assertions — are considered semantically the same, even if they have vastly different running times or resource requirements[1].

This approach was invented by Robert Floyd (1967), and popularised by Hoare (1969). Given assertions P and Q and program fragment S, define a (Floyd-)Hoare triple to be such that $\{P\}\ S\ \{Q\}$ is the assertion that if S is executed beginning with a state where P holds, then once S finishes executing, Q holds. In this case, we describe P as a precondition and Q as a postcondition. Such statements form the basis of Hoare logic, which we can use as program logic; following this, we give an axiomatic semantics for the `while` construct commonly used in imperative programs:

$$\frac{\{I \wedge G\}\ B\ \{I\}}{\{I\}\ \texttt{while}\ G\ \texttt{do}\ B\ \texttt{end}\ \{I \wedge \neg G\}}$$

This compactly states that if loop body B preserves the loop invariant I when the loop guard G is true, then we may infer that `while` G `do` B `end` when executed from an initial state where the loop invariant is true will (assuming termination) finish in a state where the invariant is still true, but also the loop guard is false. For more on axiomatic semantics, consult Slonneger & Kurtz (1995).

Operational semantics is about modelling the execution itself of a program. States of execution are mapped onto some abstract formalism, for example an automaton, which then interprets the program as a valid sequence of computation steps. Examples of this approach date back to

[1] Such concerns fall under the classification of *computational dynamics* ; most forms of semantics make identifications between notionally equivalent expressions even if they contain different amounts of computational content, in the same way that most mathematicians identify 2 + 2 with 4.

McCarthy (1960), who used the λ-calculus[2] to give an operational interpretation of the LISP programming language. Another instance of this approach is the derivation of a Control Environment (K)ontinuation (CEK) machine (Felleisen & Friedman 1986) by the application of defunctionalisation to the continuation-passing-style transformation of a definitional interpreter — such a formalism provides an abstract stack machine which interprets the language of the definitional interpreter (Reynolds 1972). An introduction to the topic of operational semantics is given in Hennessy (1990), and also in a lot more detail in Pierce (2002).

Denotational semantics is the method most relevant to this dissertation. Originally called 'mathematical semantics' or 'Scott-Strachey semantics', this approach revolves around finding mathematical objects (*denotations*) $[\![E]\!]$ to associate to expressions E of the language[3]. A collection of denotations is called a *semantic domain* .

Compositionality is key, and the denotation of any expression should in some way be the composite of the denotations of its subexpressions. Considering Listing 1, we might concoct the following denotational semantics to match our intuition:

$$
\begin{aligned}
[\![\texttt{nat}]\!] &= \mathbb{N} && \texttt{nat} \text{ type modelled by the natural numbers;}\\
[\![\texttt{0 : nat}]\!] &= [\![\texttt{0}]\!] \in [\![\texttt{nat}]\!] = 0 \in \mathbb{N} && \text{each term of type } \texttt{nat} \text{ is modelled by}\\
[\![\texttt{1 : nat}]\!] &= [\![\texttt{1}]\!] \in [\![\texttt{nat}]\!] = 1 \in \mathbb{N} && \text{its corresponding natural number;}\\
[\![\texttt{* : nat → nat → nat}]\!] &= [\![\texttt{*}]\!] \in [\![\texttt{nat → nat → nat}]\!]\\
&= [\![\texttt{*}]\!] \in [\![\texttt{nat}]\!] \Rightarrow [\![\texttt{nat → nat}]\!]\\
&= [\![\texttt{*}]\!] \in \mathbb{N} \Rightarrow [\![\texttt{nat}]\!] \Rightarrow [\![\texttt{nat}]\!] && \texttt{*} \text{ modelled by multiplication}\\
&= \times \in \mathbb{N} \Rightarrow \mathbb{N} \Rightarrow \mathbb{N} && \times \text{ on naturals;}\\
[\![\texttt{- : nat → nat → nat}]\!] &= - \in \mathbb{N} \Rightarrow \mathbb{N} \Rightarrow \mathbb{N} && \text{and similarly for subtraction;}
\end{aligned}
$$

but now we get stuck as `factorial` is an impredicative definition (it uses itself in its definition), so it isn't clear how to proceed with our naïve set-theoretic approach. Moreover, the view that terms should be interpreted by set-theoretic functions is not quite precise: let `nat` be the only base type, interpreted by the set $\mathbb{N}$ of natural numbers; then the natural interpretation for the type `nat → nat` is the function space $\mathbb{N} \Rightarrow \mathbb{N}$ — however, $\mathbb{N} \Rightarrow \mathbb{N}$ is uncountably infinite yet we can only construct countably many terms of type `nat → nat`, so there necessarily exists elements in our model that are not denotations of any term. Famously, Dana Scott solved both problems with the introduction of Scott domains (directed-complete partially ordered sets) to interpret types and (Scott-)continuous functions to interpret terms (1970).

Another key concept of this approach is the independence of semantics from syntax. To paraphrase Stoy (1977 pp. 9–10), the λ-calculus is really just machinery for the syntactic manipulation of symbols with no meaning; to solve any real problem, a semantic interpretation is necessary (as above), and it is nontrivial to prove that such an approach 'makes sense'.

In particular, we focus on **categorical semantics**, which is a generalisation of denotational semantics to the semantic domain of categories.

1.2 The substance of computer programs

Previously, we have been discussing 'computer programs' as an abstract notion, without a handle which we can grasp. Over decades of development of computability theory in the 20th century, it became

[2] At that time, the crisis of 'what is the mathematical meaning of the λ-calculus' had been somewhat resolved.

[3] The 'semantic brackets' notation $[\![-]\!]$ is commonly used in the wider literature to signify the denotation of an expression.

known that a function is partial recursive if and only if it is λ-definable (Turing 1937a) if and only if it is Turing computable (Turing 1937b), culminating in the Church-Turing thesis:

> *the informal notion of 'effectively calculable', i.e. that which can be calculated by some device or algorithm, coincides exactly with the partial recursive functions.*

We will refer to such a class of functions as *computable*.

The 'nicest class' of computer programs are arguably those that are referentially transparent, as they admit this compositionality property rather easily, in the sense that Stoy (1977 p. 5) describes as

1. the importance of an expression is solely its value,
2. subexpressions can be replaced by any expression of equal value,
3. generally, the value of a fixed expression is the same at each evaluation.

These observations parallel general mathematics, and allow for equational reasoning[4]. Such programs are described as 'functional', to wit simple programs are functions[5], which we compose to construct large programs (ultimately, bigger and more complex functions). Through the work of Plotkin (1977), we came to understand the (simply-typed) λ-calculus as the prototypical functional programming language, with Programming Computable Functions (PCF) bridging the gap between it and modern real-world functional programming languages like Haskell and the ML-family of languages.

Fig. 1. The relationship between the simply-typed λ-calculus, PCF, and Haskell.

λ-calculus is a richly developed field of study in its own right, which the classic text Barendregt's '*The Lambda Calculus: Its Syntax and Semantics*' illustrates (1984). For our purposes, it is sufficient to treat the (simply-typed) λ-calculus as a simplified notion of what is a 'computer program'.

1.3 This project

Oxford has been somewhat central in many of the developments of the formal semantics of programming languages that we have discussed, and this is reflected in modern times by its teaching. Many of the courses offered by the Department of Computer Science touch on key themes in this dissertation; roughly ordered by complexity, the following courses provide useful material to our discussion:

Functional Programming — an introductory course covering equational reasoning with Haskell, establishing basic mathematical techniques like structural induction.

Imperative Programming I — an introductory course covering axiomatic semantics by using invariants to reason about the correctness of imperative programs.

Principles of Programming Languages — a course on the operational semantics of programming languages, following Reynolds's seminal paper '*Definitional interpreters for higher-order programming languages*' (1972). Other themes include the use of monads (as equational structure) to encapsulate computational effects, like memory or continuation-passing.

Lambda Calculus and Types — a course on the λ-calculus, its developments to the fields of term-rewriting, computability theory, combinatory logic, and type theory. Many key results of the course are presented in Section 2.3.

Categories, Proofs, and Processes — a first course on category theory, and a tour of some of its applications to computer science. Also introduces the Curry-Howard-Lambek correspondence.

[4]Mathematicians often work in the general setting of first-order logic with equality where equational reasoning can be taken for granted, sidestepping any philosophical considerations.

[5]In the sense of a deterministic well-defined mapping from inputs to outputs rather than a set-theoretic notion.

Further category theory is introduced as needed.

1.3.1 Contribution and outline. The main contribution of this dissertation is the full development of the central ideas of soundness and completeness of categorical semantics for the simply-typed λ-calculus, largely following Crole (1993 Chapter 4). We present an entirely formal and rigorous approach, with no proofs omitted. In particular, our construction for the classifying category is simpler, being based on closed terms rather than terms-in-contexts, where the context is a singleton — this construction is cleaner, albeit at the expense of introducing some currying into the proofs regarding it. In addition, we pay meticulous care to the consideration of non-strict Cartesian closed/product-preserving functors, especially in the discussion of results about model-translating functors (Definition 4.14).

This is done in two parts: we first establish the syntax we are working with in Section 2, in the form of $\lambda^{\to\times}$-theories. Then, in Section 3, we present a semantics for such theories in the form of Cartesian closed category. Section 4 is the main technical content of this dissertation, split into covering soundness in Section 4.1, and completeness in Section 4.2. Completeness requires much more machinery to adequately develop: first, we talk about syntactic categories which are freely constructed by $\lambda^{\to\times}$-theories (Section 4.2.1); then we discuss how the collection categorical models of a $\lambda^{\to\times}$-theory themselves have categorical structure (Section 4.2.2); in Section 4.2.3, we combine the machinery together to establish the completeness result.

The reader should keep in mind the question of 'to what extent are simply-typed λ-calculi *the same* as Cartesian closed categories' throughout, as this is our central theme.

TEXcount word count: 9753

2 λ-CALCULUS

In this section, we develop the λ-calculus necessary to give a syntax to talk about programs. An appropriate collection of this syntax will be the subject of our semantic modelling.

The λ-calculus was invented by Alonzo Church to be a formal system for the foundations of logic, centred around the primitive notion of function rather than set, first published in (1932). Somewhat serendipitously, the notion of *λ-definable function* turned out to be extremely expressive, and this ultimately lead to the Church's famous thesis in computability (Section 1.2). Another great achievement of the era was the first proof of a negative result for Hilbert's *Entscheidungsproblem*[6], opening up the realm of the undecidable (Church 1936).

In the 1950s–1960s, the λ-calculus began to draw the attention of computer scientists, notably influencing John McCarthy's LISP (1960). Moreover, the idea that λ-calculus could be *the* prototypical (functional) programming language was further developed by Böhm, resulting in the CUCH programming language, which combined CUrry's combinatory logic with CHurch's λ-calculus (1966).

The history and relevance of the λ-calculus is an enormous topic; far too much to recount in this project — an excellent review is contained in Cardone & Hindley (2006).

2.1 Syntax

We formalise the λ-calculus as a mathematical *formal theory*.

Definition 2.1 ($\lambda^{\to\times}$-signature). A $\lambda^{\to\times}$-signature, σ, is given by the following data:

- Fix a set TV of *ground types*. The collection of *simple types* over TV, $\mathsf{ST}(\mathsf{TV})$, is defined by the following BNF:

$$\mathsf{ST}(\mathsf{TV}): \quad A ::= G \mid \mathsf{unit} \mid A \to A \mid A \times A \quad \text{where } G \in \mathsf{TV}.$$

- A collection of *function symbols*, each with arity a natural number. If f is a n-ary function symbol, such that $n > 0$, it is associated to a mapping $f\colon A_1 \times \cdots \times A_n \to B$. A 0-ary function symbol c is associated to a type, $c{:}B$, and this is described as the *constant* c of type B. We use c to mean $c()$ in the context of function application when it is clear from context.

Definition 2.2 (λ-terms). The collection of *raw terms* generated by σ, Λ, assuming a countably infinite set $\mathcal{V}$ of variables, is defined as:

$$\Lambda: \quad t ::= x \mid \underbrace{f(t, \dots, t)}_{n \text{ times}} \mid \lambda x{:}A.t \mid tt \mid \langle\rangle \mid \langle t, t\rangle \mid \mathsf{fst}(t) \mid \mathsf{snd}(t),$$

where $x \in \mathcal{V}$, and f is a n-ary function symbol.

A raw term can be thought of as a syntactically valid expression which may be ill-typed. Function symbols can be thought of as primitive functions with a meaning outside of the theory, while function types are the core construction integral to simply-typed lambda calculi. Product types can be encoded with function types, but for a nice parallel with Cartesian closed categories (preserving the categoricity of product), we include them as part of the signature.

Every variable appears either free, or bound by some λ-abstraction, and terms are identified modulo renaming of bound variables. More formally, we define the subterm relation $\subset$ over raw terms inductively.

Definition 2.3 (Subterms). For terms $s, t, u, v \in \Lambda$ and $x \in \mathcal{V}$:

[6] Given a formula of first-order logic, is it valid? That is, does every assignment of variables to truth values make the whole formula true?

$$\frac{}{t \subset t} \qquad \frac{\exists 1 \leq i \leq n . s \subset t_i}{s \subset f(t_1, \ldots, t_n)}\ f \text{ is an } n\text{-ary function symbol} \qquad \frac{s = x \vee s \subset t}{s \subset \lambda x{:}A.t}$$

$$\frac{s \subset t}{s \subset \mathsf{fst}\,(t)} \qquad \frac{s \subset t}{s \subset \mathsf{snd}\,(t)} \qquad \frac{t \subset u \vee t \subset v}{t \subset uv} \qquad \frac{t \subset u \vee t \subset v}{t \subset \langle u, v \rangle}$$

Notice that constants and $\langle\rangle$ are atomic.

Now, for subterms of the form $\lambda x{:}A.t$, we describe t as the *scope* of the binding $\lambda x{:}A$ and x as a *bound* variable in scope. If x was already bound, then it is rebound in the tightest enclosing scope and this is described as *variable capture*.

Example 2.4. In the term

$$\lambda x{:}A.(\lambda x{:}A.vx),$$

x has been captured by the innermost λ-abstraction, and is a distinct variable to x. A term with equal 'meaning' (but not identical syntax) is $\lambda x{:}A.(\lambda z{:}A.vz)$, and we seek to establish an identification of the two.

Free variables are the variables which do not appear bound at the top level (largest scope) of the raw term, and can be given inductively as follows.

Definition 2.5 (Free variables). The set of free variables of a term t, $\mathrm{fv}\,(t)$, is defined inductively:

$$\begin{aligned} \mathrm{fv}\,(c) = \mathrm{fv}\,(\langle\rangle) &:= \varnothing, && c \text{ is a constant}, \\ \mathrm{fv}\,(x) &:= \{x\}, && x \in \mathcal{V}, \\ \mathrm{fv}\,(uv) = \mathrm{fv}\,(\langle u, v \rangle) &:= \mathrm{fv}\,(u) \cup \mathrm{fv}\,(v), \\ \mathrm{fv}\,(\lambda x{:}A.t) &:= \mathrm{fv}\,(t) \setminus \{x\}, \\ \mathrm{fv}\left(\mathsf{fst}\,(t)\right) = \mathrm{fv}\left(\mathsf{snd}\,(t)\right) &:= \mathrm{fv}\,(t). \end{aligned}$$

Example 2.6. In the term

$$t = (\underbrace{\lambda x{:}A}_{\dagger} . \underbrace{ux}_{*})(yx), \qquad x, y \in \mathcal{V}$$

y appears free, but x appears both bound (left and middle occurrences) and free (right occurrence). Moreover, x appearing in the scope $(*)$ is captured by the innermost λ-abstraction $(\dagger)$. Variable capture is problematic when we come to define substitution, so we first formalise a notion of relating terms which are not syntactically equal, but have the same 'meaning'.

Definition 2.7 (α-equivalence). We define a relation on raw terms $\alpha_=$ inductively

$$\frac{}{x =_\alpha x}\ x \in \mathcal{V} \qquad \frac{}{c =_\alpha c}\ c \text{ is a constant}$$

$$\frac{t_1 =_\alpha t_1' \qquad \ldots \qquad t_n =_\alpha t_n'}{f(t_1, \ldots, f_n) =_\alpha f(t_1', \ldots, t_n')}\ f \text{ is a } n\text{-ary function symbol}$$

$$\frac{u =_\alpha u' \qquad v =_\alpha v'}{uv =_\alpha u'v'}$$

$$\frac{t[y/x] =_\alpha t'[y/x']}{\lambda x{:}A.t =_\alpha \lambda x'{:}A.t'}\ x, y \in \mathcal{V}, \text{ and } y \text{ does not appear in } tt'$$

$$\frac{}{\langle\rangle =_\alpha \langle\rangle} \qquad \frac{u =_\alpha u' \qquad v =_\alpha v'}{\langle u, v \rangle =_\alpha \langle u', v' \rangle} \qquad \frac{t =_\alpha t'}{\mathsf{fst}\,(t) =_\alpha \mathsf{fst}\,(t')} \qquad \frac{t =_\alpha t'}{\mathsf{snd}\,(t) =_\alpha \mathsf{snd}\,(t')}$$

α-equivalence can be thought of as the renaming of bound variables preserving semantics, and will allow us to conveniently ignore concerns regarding variable capture (as we can always α-convert to an equivalent term for which variable capture would not occur, and $\mathcal{V}$ is countably infinite so we do not 'run out' of variables). This forms an equivalence class over Λ, and we shall refer to the representatives of elements of $\Lambda/_{=_\alpha}$ as *terms*. In doing this, we emphasise that we do not distinguish between α-equivalent raw terms.

Definition 2.8 (Substitution). For a term t and a variable x, substitution is defined inductively:

$$\begin{aligned} y[t/x] &\coloneqq \begin{cases} t, & \text{if } y = x \\ y, & \text{if } y \neq x \end{cases} && y \in \mathcal{V} \\ c[t/x] &\coloneqq c, && c \text{ is a constant} \\ f(t_1, \dots, t_n)[t/x] &\coloneqq f(t_i[t/x], \dots, t_n[t/x]), && f \text{ is a } n\text{-ary function symbol} \end{aligned}$$

(function types)

$$\begin{aligned} (\lambda z{:}A.u)[t/x] &\coloneqq \lambda z{:}A.u[t/x], && (\star) \\ (uv)[t/x] &\coloneqq u[t/x]v[t/x], \end{aligned}$$

(product types)

$$\begin{aligned} \langle\rangle[t/x] &\coloneqq \langle\rangle, & \langle u, v\rangle[t/x] &\coloneqq \langle u[t/x]v[t/x]\rangle, \\ \mathsf{fst}\,(u)\,[t/x] &\coloneqq \mathsf{fst}\left(u[t/x]\right), & \mathsf{snd}\,(u)\,[t/x] &\coloneqq \mathsf{snd}\left(u[t/x]\right). \end{aligned}$$

($\star$) is the condition that $z \notin \mathrm{fv}\,(t) \cup \mathrm{fv}\,(x)$, and can always be satisfied by α-converting z to some 'fresh' variable.

Definition 2.9 (β-equivalence). Evaluation is formalised by *β-reduction*, a relation $\to_\beta$ defined by

$$\begin{aligned} (\lambda x{:}A.t)u &\to_\beta t[u/x], \\ \mathsf{fst}\,(\langle u, v\rangle) &\to_\beta u, \\ \mathsf{snd}\,(\langle u, v\rangle) &\to_\beta v, \end{aligned}$$

and we write $\twoheadrightarrow_\beta$ to be the reflexive transitive closure induced by $\to_\beta$. For terms u and v such that $u \twoheadrightarrow_\beta v$, we say that the *redex* u *reduces* to the *contractum* v, and $\twoheadrightarrow_\beta$ forms a partial order. We take *β-equivalence* to be the identification $u =_\beta v$, whereby there is some t such that $u \twoheadrightarrow_\beta t$ and $v \twoheadrightarrow_\beta t$.

β-reduction gives a notion of computational dynamics. By this, we mean that β-equivalent terms have the 'same meaning', but different amounts of computational content in them. For example, in ordinary arithmetic, we consider $2 + 2$ and 4 to be equal, however the former has more computational content than the latter because we establish the equality by 'evaluating' the addition. λ-calculus as an abstract theory of functions precisely captures this notion of evaluation with β-reduction and substitutions.

Definition 2.10 (η-equivalence). Similarly to β-reduction, *η-reduction* is a relation $\to_\eta$ defined by

$$\begin{aligned} \lambda x{:}A.tx &\to_\eta t, \\ \langle \mathsf{fst}\,(v)\,, \mathsf{snd}\,(v)\rangle &\to_\eta v, \end{aligned}$$

from which a reflexive transitive closure $\twoheadrightarrow_\eta$ is induced. Like before, we take *η-equivalence* to be the identification $u =_\eta v$, adding symmetry.

η-reduction formalises the notion that the only thing to be done with a lambda term is application: consider the reduction

$$(\lambda x{:}A.ux)v \twoheadrightarrow uv,$$

which can reduce via both β or η — η-reduction would 'anticipate' the β redex that arises when the lambda term containing u is applied to v. η-equivalence can give an identification to *observationally equivalent* terms[7], such as $\lambda y{:}A.\lambda x{:}A.yx =_\eta \lambda x{:}A.x$. In a sense, η-reduction is a special case of the extensionality of functions in our λ-calculus.

Definition 2.11 ($\beta\eta$-equivalence). *$\beta\eta$-reduction* $\rightsquigarrow$ is the transitive closure $\rightsquigarrow_\beta \cup \rightsquigarrow_\eta$, and *$\beta\eta$-equivalence* $=_{\beta\eta}$ is the transitive closure $=_\beta \cup =_\eta$.

$\beta\eta$-equivalence is the canonical notion of equivalence for lambda terms we consider, capturing equality modulo computational dynamics. This canonicity is categorical in the sense that $\beta\eta$-equivalent terms are identified as equal morphisms in the classifying category, which will be shown later.

2.2 Typed terms

Previously, we have only been considering unityped terms of type A, without much care, which describes something equivalent to the untyped λ-calculus (Harper 2016 sec. 21.4) — see also Example 2.34.

The untyped λ-calculus has properties which make it inelegant to model, such as the undecidability of term equality, emerging due to logical inconsistency when viewed as a logic, as shown by the Kleene-Rosser paradox (1935). For an example of this sort, consider the following.

Example 2.12 (Russell's paradox in untyped λ-calculus). Let not be the λ-term corresponding to logical negation, i.e. if λ-terms are assigned truth values invariant under β-equivalence, then $\mathsf{not}(t)$ is assigned the negation of the truth value of t. Observe that $\mathbf{y} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ is a fixed-point combinator: a closed λ-term such that for all terms t,

$$\mathbf{y}t =_\beta t(\mathbf{y}t).$$

Then we can derive Russell's paradox:

$$\mathbf{y}\mathsf{not} =_\beta \mathsf{not}(\mathbf{y}\mathsf{not}).$$

$\mathbf{y}\mathsf{not}$ is assigned a truth value as it is a term, but it is β-equivalent to $\mathsf{not}(\mathbf{y}\mathsf{not})$ which should be assigned the same truth value, but due to assumption on not it should be assigned the opposite truth value — paradox.

One method to resolve this is the introduction of the *simply-typed λ-calculus*, formulated by Church (1940) and built on the work of Russell and Whitehead, which restricts the constructible terms to a 'well-behaved' strict subset. For instance, no simply-typed theory of the λ-calculus admits fixed-point combinators.

Types impose a restriction on reduction that prevents pathological behaviour, like self-application. For instance, a term of type $A \to B$ may only be applied to terms of type A, eliding the possibility of ever applying terms which are not morally 'functions'. Viewing the λ-calculus as an abstract programming language, the enforcement of well-typing elides the possibility of writing nonsensical (i.e. ill-typed) terms.

The presentation of the simply-typed λ-calculus here includes additional rules for product types and terms, which preserve the categoricity of Cartesian product in the model.

We talk about typed terms in relation to a context, defined as follows.

Definition 2.13 (Typing contexts). A *typing context* is defined by a finite list of pairs of (variable, type)

$$\Gamma \coloneqq [x_1{:}A_1, \dots, x_n{:}A_n], \quad (x_1, \dots, x_n \text{ are all distinct})$$

where $x_1, \dots, x_n \in \mathcal{V}$, $A_1, \dots, A_n \in \mathsf{ST}\,(\mathsf{TV})$.

[7] Two terms u and v terms are observationally equivalent if and only if for all terms t, (tu terminates $\iff$ tv terminates), where a term terminates if and only if it cannot reduce infinitely.

A typing context Γ (sometimes called an *environment*) *binds* types to free variables in a term, written $\sigma \triangleright \Gamma \vdash t$. We call $\sigma \triangleright \Gamma \vdash t$ a *term-in-context*, and often we omit $\sigma\triangleright$ when the signature is clear. We write + for list append.

Definition 2.14 (Typing relation). We have the following inductive rules for forming types:

$$\text{var}\ \frac{}{\Gamma + [x{:}A] + \Gamma' \vdash x : A} \qquad \text{unit}\ \frac{}{\Gamma \vdash \langle\rangle : \mathsf{unit}}$$

$$\text{const}\ \frac{}{\Gamma \vdash c : A}\ c \text{ is a constant of type } A$$

$$\text{func}\ \frac{\Gamma \vdash t_1 : A_1 \qquad \dots \qquad \Gamma \vdash t_n : A_n}{\Gamma \vdash f(t_1, \dots, t_n) : B}\ f \text{ is an } n\text{-ary function symbol of type } A_1 \times \cdots \times A_n \to B$$

$$\text{abs}\ \frac{\Gamma + [x{:}A] + \Gamma' \vdash t : B}{\Gamma + \Gamma' \vdash (\lambda x : A.t) : A \to B} \qquad \text{app}\ \frac{\Gamma \vdash u : A \to B \qquad \Gamma \vdash v : A}{\Gamma \vdash uv : B}$$

$$\text{pair}\ \frac{\Gamma \vdash u : A \qquad \Gamma \vdash v : B}{\Gamma \vdash \langle u, v\rangle : A \times B}$$

$$\mathsf{fst}\ \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathsf{fst}\,(t) : A} \qquad \mathsf{snd}\ \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathsf{snd}\,(t) : B}$$

The typing relation imposes a *judgement* on whether or not a term is well-typed.

Proposition 2.15 (Exchange). *Every permutation of typing contexts is equivalent:*

$$\mathit{exchange}\ \frac{\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash t : C}{\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash t : C}$$

Proof. By induction on the derivation of $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash t : C$.

var case By assumption, we have that $t{:}C = x{:}A$, or $t{:}C = y{:}B$, or some other element of the list $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3$; in any case, $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash t : C$ is immediately derivable as an instance of var. ◁

const case By assumption, we have that $t{:}C$ is a constant, and so $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash t : C$ is immediately derivable as an instance of const. ◁

app case By assumption, we have that $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash u : A \to C$ and $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash v : A$ such that $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash uv = t : C$. The inductive hypothesis states that $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash u : A \to C$ and $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash v : A$, from which we can apply app to derive $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash uv = t : C$. ◁

fst, snd case By assumption, we have that $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash u : C \times B$ such that $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash \mathsf{fst}\,(u) = t : C$. The inductive hypothesis states that $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash u : C \times B$, from which we can apply fst to derive $\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash \mathsf{fst}\,(v) = t : C$. The case for **snd** is similar. ◁

□

Proposition 2.16 (Weakening). *Enlarging a typing context does not invalidate its typing judgements:*

$$\mathit{weaken}\ \frac{\Gamma \vdash t : A}{\Gamma + [x{:}B] \vdash t : A}\ x \textit{ does not occur in } \Gamma$$

Proof. By induction over the derivation of $\Gamma \vdash t : A$. Almost identical to proof of Proposition 2.15. □

Proposition 2.17 (Substitution). *Any free variable can be swapped with a term of equal type while maintaining typing judgement:*

$$\textit{subst}\ \frac{\Gamma + [x{:}A] + \Gamma' \vdash v : B \qquad \Gamma + \Gamma' \vdash u : A}{\Gamma + \Gamma' \vdash v[u/x] : B}$$

Proof. By induction on the derivation of $\Gamma + [x{:}A] + \Gamma' \vdash v : B$.

var case By assumption, we have that $v{:}B = x{:}A$, or $v{:}B$ is some element of the list $\Gamma + \Gamma'$; in the first case, we have $B = A$ and $v[u/x] = x[u/x] = u$ and so $\Gamma + \Gamma' \vdash v[u/x] : A$. Otherwise, $v \neq x$ is some variable (as x may only occur once in the list $\Gamma + [x{:}A] + \Gamma'$), so $\Gamma + \Gamma' \vdash v : B$ as a result of applying var, and the substitution $[u/x]$ has no effect, hence $\Gamma + \Gamma' \vdash v = v[u/x] : B$. ◁

const case By assumption, we have that $v{:}B$ is a constant, and so the substitution $[u/v]$ has no effect, allowing us to immediately derive $\Gamma + \Gamma' \vdash v = v[u/x] : B$ as an instance of const. ◁

app case By assumption, we have that $\Gamma + [x{:}A] + \Gamma' \vdash p : C \to B$ and $\Gamma + [x{:}A] + \Gamma' \vdash q : C$ such that $\Gamma + [x{:}A] + \Gamma' \vdash pq = v : B$. By the inductive hypothesis, we derive $\Gamma + \Gamma' \vdash p[u/x] : C \to B$ and $\Gamma + \Gamma' \vdash q[u/x] : C$, from which we can derive $\Gamma + \Gamma' \vdash p[u/x]q[u/x] = (pq)[u/x] = v[u/x] : B$ as an instance of app. ◁

fst, snd case By assumption, we have that $\Gamma + [x{:}A] + \Gamma' \vdash t : B \times C$ such that $\Gamma + [x{:}A] + \Gamma' \vdash \mathsf{fst}\,(t) = v : B$. The inductive hypothesis states that $\Gamma + \Gamma' \vdash t[u/x] : B \times C$, from which we can apply fst to derive $\Gamma + \Gamma' \vdash \mathsf{fst}\left(t[u/x]\right) = \mathsf{fst}\,(t)\,[u/x] = v[u/x] : B$. The case for **snd** is similar. ◁

□

Theorem 2.18 (Type uniqueness). *If* Γ *is a typing context, and* t *is a term such that* $\Gamma \vdash t : A$ *and* $\Gamma \vdash t : B$*, then* $A = B$*.*

Proof. By induction on the derivation of $\Gamma \vdash t : A$. Almost identical to proof of Proposition 2.15. □

Therefore, under a typing context, if a term has a type, the type is unique, and we call such a term *typable*. However, consider the two distinct typing judgement derivations for $\Gamma + [x{:}A] + [y{:}B] \vdash t : C$ supposing we already have a fixed derivation for $\Gamma \vdash t : C$:

$$\textsf{weaken}\ \frac{\textsf{weaken}\ \dfrac{\dfrac{\ldots}{\Gamma \vdash t : C}}{\Gamma + [x{:}A] \vdash t : C}}{\Gamma + [x{:}A] + [y{:}B] \vdash t : C} \qquad \textsf{weaken}\ \frac{\textsf{weaken}\ \dfrac{\dfrac{\ldots}{\Gamma \vdash t : C}}{\Gamma + [y{:}B] \vdash t : C}}{\textsf{exchange}\ \dfrac{\Gamma + [y{:}B] + [x{:}A] \vdash t : C}{\Gamma + [x{:}A] + [y{:}B] \vdash t : C}}$$

In general, the typing judgement derivation for a term is not unique! But this is not problematic as we could recover uniqueness by being extremely fastidious about syntactic identifications. The technique of de Bruijn indices (Bruijn 1972) allows us to represent the syntax of the λ-calculus in such a way that terms are invariant under α-equivalence, so two terms are considered α-equivalent if and only if they are syntactically equal — essentially, variables bound in a context are replaced with indices fully determined by the structure of the term; then, we only accept typing contexts with subjects which are numerically ordered.

Ben-Yelles (1979) showed that with reasonable constraints on the typing context, we have the following result:

Lemma 2.19 (Deduction uniqueness for β-normal terms). *If* t *is a term in* β*-normal form, with a deduction*[8] Δ *of* $\Gamma \vdash t : A$*, then*

(i) *every type in* Δ *has an occurrence in* A *or in a type in* Γ*,*
(ii) Δ *is unique.*

[8] A *deduction* is a restricted form of derivation, as a mathematical object — see Hindley (1997 p. 16).

Proof. See Hindley (1997 Chapter 2B). □

This can be extended to the syntax we employ, for some 'reasonable constraints' (e.g. insisting on binary products only, and asserting that $\langle\langle x, y\rangle, z\rangle \neq \langle x, \langle y, z\rangle\rangle$); however, these concerns are entirely bureaucratic and we shall not consider them.

These type-forming rules define how to assign *simple-types* to terms, and we will concentrate only terms which have a simple-type (i.e. they are typable). This λ-calculus is more restricted, but has some properties which make it nicer to work with, which we discuss next.

2.3 Properties of λ-calculi

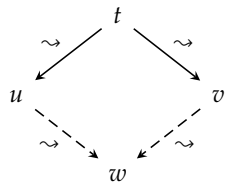
In this section, we briefly outline some key properties of λ-calculi.

Definition 2.20 (Consistency). A λ-calculus is *consistent* if there exist distinct terms which are not equated.

It seems like a very basic requirement that any λ-calculus we consider is consistent, else such a theory cannot distinguish between any of its terms. The introduction of too many equalities between terms in the theory can cause it to collapse into inconsistency.

Definition 2.21 (Church-Rosser). Take $\twoheadleftarrow$ to be the symmetric reflexive transitive closure of a reduction relation $\rightarrow$. A λ-calculus has the *Church-Rosser property* if, for terms x and y such that $x \twoheadleftarrow y$, there exists some term z such that $x \rightsquigarrow z$ and $y \rightsquigarrow z$.

Definition 2.22 (Confluence). Term reduction in a λ-calculus is *confluent* if, given a term t such that $t \rightsquigarrow u$ and $t \rightsquigarrow v$, there exists a term w such that $u \rightsquigarrow w$ and $v \rightsquigarrow w$:



This is the *diamond property*, and it states that different reduction orders will not yield contravening terms. Confluence is equivalent to the Church-Rosser property (O'donnell 1977 Chapter 4), and the two terms are used interchangeably in the literature. For a comprehensive text on term rewriting, consult Baader & Nipkow (1998).

Theorem 2.23. *$\beta\eta$-reduction in $\lambda^{\rightarrow\times}$ is confluent.*

Proof. See Barendregt (1984 p. 67).

□

Confluence allows us to prove the following desirable property about terms.

Corollary 2.24. *All normal forms of terms in $\lambda^{\rightarrow\times}$ are unique.*

Proof. Suppose that u and v are normal forms of t, i.e. $t \rightsquigarrow u$ and $t \rightsquigarrow v$. Then by confluence, there is some w such that $u \rightsquigarrow w$ and $v \rightsquigarrow w$. But u and v are normal, so can only reduce to themselves, hence $u = w = v$.

□

The consistency of $\lambda^{\rightarrow\times}$ immediately follows.

Theorem 2.25 (Weak normalisation). *There exists a reduction strategy for the normalisation of terms in $\lambda^{\rightarrow\times}$ which terminates.*

Proof. See Girard *et al.* (1989 pp. 24–25).

□

This result allows us to computationally discern term equality.

Corollary 2.26 (Decidability of term equivalence). *The equivalence of two terms u and v in $\lambda^{\to\times}$ is decidable.*

Proof. As $\lambda^{\to\times}$ is weakly normalising, we have a terminating strategy to normalise both u and v. $u = v$ if and only if they have the same normal form.

□

However, for the simply-typed λ-calculus an even stronger result holds.

Theorem 2.27 (Strong normalisation). *Every reduction strategy for the normalisation of terms in $\lambda^{\to\times}$ terminates.*

Proof. See Girard *et al.* (1989 Chapter 6).

□

These results show that a normal form is computable for any typable term, which underlies the notion that typable terms are 'well-behaved', and that such a normal form is necessarily unique; furthermore, *every* reduction brings us closer to a normal form, and there is no way to reduce infinitely.

2.4 $\lambda^{\to\times}$-theories

Now we have established the basic machinery of λ-calculus, we can discuss a *$\lambda^{\to\times}$-theory* (in the sense of a collection of terms closed under $\beta\eta$-equality). This is the precise mathematical object to which a categorical semantics can be attributed.

Definition 2.28 ($\lambda^{\to\times}$-axiom). A *$\lambda^{\to\times}$-axiom* is an *equation-in-context* of the form

$$\sigma \rhd \Gamma \vdash u = v : A,$$

which is an identification between $\Gamma \vdash u : A$ and $\Gamma \vdash v : A$ internalised in the theory.

Definition 2.29 ($\lambda^{\to\times}$-theory). A $\lambda^{\to\times}$-theory, $\mathcal{T}$, is a signature σ accompanied by a collection of axioms $\mathcal{A}$

$$\mathcal{T} \coloneqq (\sigma, \mathcal{A}).$$

The *theorems* of $\mathcal{T}$ are the equations-in-context *derivable* from $\mathcal{A}$ under signature σ. This notion of derivability is defined precisely as

$$\text{axiom}\ \frac{\mathcal{A} \rhd \Gamma \vdash u = v : A}{\mathcal{T} \rhd \Gamma \vdash u = v : A}$$

$$\text{refl}\ \frac{\sigma \rhd \Gamma \vdash t : A}{\mathcal{T} \rhd \Gamma \vdash t = t : A} \qquad \text{sym}\ \frac{\mathcal{T} \rhd \Gamma \vdash u = v : A}{\mathcal{T} \rhd \Gamma \vdash v = u : A}$$

$$\text{trans}\ \frac{\mathcal{T} \rhd \Gamma \vdash t = u : A \qquad \mathcal{T} \rhd \Gamma \vdash u = v : A}{\mathcal{T} \rhd \Gamma \vdash t = v : A}$$

$$\text{unit}\ \frac{\sigma \rhd \Gamma \vdash t : \mathsf{unit}}{\mathcal{T} \rhd \Gamma \vdash t = \langle\rangle : \mathsf{unit}}$$

$$\beta\ \frac{\sigma \rhd \Gamma + [x{:}A] \vdash u : B \qquad \sigma \rhd \Gamma \vdash v : A}{\mathcal{T} \rhd \Gamma \vdash (\lambda x{:}A.u)v = u[v/x] : B} \qquad \eta\ \frac{\sigma \rhd \Gamma \vdash t : A \to B}{\mathcal{T} \rhd \Gamma \vdash (\lambda x{:}A.tx) = t : A \to B}\ x \notin \mathrm{fv}(t)$$

$$\text{abs}\ \frac{\mathcal{T} \rhd \Gamma + [x{:}A] \vdash u = v : B}{\mathcal{T} \rhd \Gamma \vdash \lambda x{:}A.u = \lambda x{:}A.v : A \to B} \qquad \text{app}\ \frac{\mathcal{T} \rhd \Gamma \vdash s = t : A \to B \qquad \mathcal{T} \rhd \Gamma \vdash u = v : A}{\mathcal{T} \rhd \Gamma \vdash su = tv : B}$$

$$\text{pair}\ \frac{\sigma \rhd \Gamma \vdash t : A \times B}{\mathcal{T} \rhd \Gamma \vdash \langle \mathsf{fst}\,(t)\,,\,\mathsf{snd}\,(t)\rangle = t : A \times B}$$

$$\mathsf{fst}\ \frac{\sigma \rhd \Gamma \vdash u : A \qquad \sigma \rhd \Gamma \vdash v : B}{\mathcal{T} \rhd \Gamma \vdash \mathsf{fst}\,(\langle u, v\rangle) = u : A} \qquad \mathsf{snd}\ \frac{\sigma \rhd \Gamma \vdash u : A \qquad \sigma \rhd \Gamma \vdash v : B}{\mathcal{T} \rhd \Gamma \vdash \mathsf{snd}\,(\langle u, v\rangle) = v : B}$$

These rules ensure that the identification $\Gamma \vdash u = v : A$ is closed under $u =_{\beta\eta} v$.

We now have a few results reflecting the results of Section 2.2 for $\lambda^{\to\times}$-theorems.

Proposition 2.30 (Exchange for derivations). *An identification made in $\mathcal{T}$ under a context will hold under any permutation of that context:*

$$\mathit{exchange}\ \frac{\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash u = v : C}{\Gamma_1 + [y{:}B] + \Gamma_2 + [x{:}A] + \Gamma_3 \vdash u = v : C}$$

Proof. Trivial induction over the derivation of $\Gamma_1 + [x{:}A] + \Gamma_2 + [y{:}B] + \Gamma_3 \vdash u = v : C$ using Proposition 2.15.

□

In the categorical semantics, this makes sense as typing contexts are modelled by finite products; two products with exactly the same data are canonically isomorphic irrespective of their ordering, and so are notionally 'the same'.

Proposition 2.31 (Weakening for derivations). *An identification made in $\mathcal{T}$ will continue to hold in a larger context:*

$$\mathit{weaken}\ \frac{\Gamma \vdash u = v : A}{\Gamma + [x{:}B] \vdash u = v : A}\ x \textit{ does not occur in } \Gamma$$

Proof. Trivial induction over the derivation of $\Gamma \vdash u = v : A$ using Proposition 2.16.

□

Categorically, one can view the model of the consequent as the projection of the the larger typing context to the smaller one composed with the model of the antecedent.

Similarly to before, we have a substitution theorem, slightly stronger than Proposition 2.17.

Proposition 2.32 (Substitution for derivations). *Equality under $\mathcal{T}$ is invariant under substituting a free variable for equal terms judged under the remaining context:*

$$\mathit{subst}\ \frac{\mathcal{T} \rhd \Gamma + [x{:}A] \vdash s = t : B \qquad \mathcal{T} \rhd \Gamma \vdash u = v : A}{\mathcal{T} \rhd \Gamma \vdash s[u/x] = t[v/x] : B}$$

Proof. By assumption, we have $\sigma \rhd \Gamma + [x{:}A] \vdash s : B$ and $\sigma \rhd \Gamma \vdash u : A$, so by β we derive $\mathcal{T} \rhd \Gamma \vdash (\lambda x{:}A.s)u = s[u/x] : B$, and similarly derive $\mathcal{T} \rhd \Gamma \vdash (\lambda x{:}A.t)v = t[v/x] : B$.

Now,

$$\dfrac{\dfrac{\text{assumption}\ \dfrac{}{\mathcal{T} \rhd \Gamma + [x{:}A] \vdash s = t : B}}{\text{abs}\ \dfrac{}{\mathcal{T} \rhd \Gamma \vdash \lambda x{:}A.s = \lambda x{:}A.t : A \to B}} \qquad \text{assumption}\ \dfrac{}{\mathcal{T} \rhd \Gamma \vdash u = v : A}}{\text{app}\ \dfrac{}{\mathcal{T} \rhd \Gamma \vdash (\lambda x{:}A.s)u = (\lambda x{:}A.t)v : B}}$$

The proof can be completed by application of equational reasoning rules.

□

We end this section with some basic but important examples adapted from Awodey & Bauer (2017 pp. 79–82) which exemplify the expressivity of $\lambda^{\to\times}$-theories.

Example 2.33 (Theory of monoids). The *theory of monoids* is a $\lambda^{\to\times}$-theory, with signature

$$\mathsf{TV} = \{X\}, \quad e{:}X, \quad \cdot{:}X \times X \to X,$$

and axioms (writing $x \cdot y$ for $\cdot(\langle x, y\rangle)$)

$$\begin{aligned}[x{:}X, y{:}X, z{:}X] &\vdash (x \cdot y) \cdot z = x \cdot (y \cdot z) : X,\\ [x{:}X] &\vdash e \cdot x = x : X,\\ [x{:}X] &\vdash x \cdot e = x : X.\end{aligned}$$

These axioms give us precisely the properties of a monoid: associativity of the monoid operation $\cdot$ and a distinguished identity $e{:}X$.

It is clear how to extend this to the theory of groups, rings, or other algebraic structures — in fact, any algebraic theory determines a $\lambda^{\to\times}$-theory.

Example 2.34 (Theory of extensional reflexive type). The unityped *theory of extensional reflexive type* is a $\lambda^{\to\times}$-theory with signature

$$\mathsf{TV} = \{T\}, \quad \mathbf{r}{:}T \to (T \to T), \quad \mathbf{s}{:}(T \to T) \to T,$$

and axioms

$$\begin{aligned}[f{:}T \to T] &\vdash \mathbf{r}(\mathbf{s}(f)) = f : T \to T,\\ [x{:}T] &\vdash \mathbf{s}(\mathbf{r}(x)) = x : T.\end{aligned}$$

These equations give an isomorphism between types

$$T \cong T \to T,$$

so this theory is really the theory of the unityped λ-calculus, and by syntactic translation represents the untyped λ-calculus: we translate an untyped term t into a unityped one t^* inductively as follows

$$\begin{aligned}x^* &= x, && \text{for } x \in \mathcal{V}\\ (uv)* &= (\mathbf{r}(u^*))v^*,\\ (\lambda x.u)^* &= \mathbf{s}(\lambda x{:}T.u^*).\end{aligned}$$

In order to talk about terms-in-context, we bootstrap a new typing context with every free variable typed as T:

$$[x_1{:}T, \dots, x_n{:}T] \vdash t^* : T,$$

where $\mathrm{fv}(t) = \{x_1, \dots, x_n\}$.

Example 2.35 (Theory of PCF). The theory of PCF is a $\lambda^{\to\times}$-theory given by signature

$$\mathsf{TV} = \{\mathtt{nat}, \mathtt{bool}\}$$

with constants and function symbols

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

for each type $A \in \mathsf{ST}(\mathsf{TV})$. The equations are generated from the axioms

$$\begin{aligned}
\vdash\ & \mathtt{iszero}(0) = \mathtt{true} : \mathtt{bool},\\
[n{:}\mathtt{nat}] \vdash\ & \mathtt{iszero}(\mathtt{succ}(n)) = \mathtt{false} : \mathtt{bool},\\
[n{:}\mathtt{nat}] \vdash\ & \mathtt{pred}(0) = 0 : \mathtt{nat},\\
[n{:}\mathtt{nat}] \vdash\ & \mathtt{succ}(\mathtt{pred}(\mathtt{succ}(n))) = \mathtt{succ}(n) = \mathtt{pred}(\mathtt{succ}(\mathtt{succ}(n))) : \mathtt{nat},\\
[t{:}A, f{:}A] \vdash\ & \mathtt{cond}_A(\mathtt{true})(t)(f) = t : A,\\
[t{:}A, f{:}A] \vdash\ & \mathtt{cond}_A(\mathtt{false})(t)(f) = f : A,\\
[f{:}A \to A] \vdash\ & \mathtt{fix}_A(f) = f(\mathtt{fix}_A(f)) : A.
\end{aligned}$$

$\mathtt{fix}_A$ is a fixed point combinator for the type A, and while we could do away with the other constants and function symbols by means of encoding (e.g. Church's), this really needs to be provided in the meta-theory.

In the theory of PCF, we can write our program from Listing 1 as

$$\begin{aligned}
\mathrm{BIN} &= \mathtt{nat} \to \mathtt{nat} \to \mathtt{nat}, \quad \mathrm{UN} = \mathtt{nat} \to \mathtt{nat},\\
F &= (\mathrm{BIN} \to \mathrm{BIN}) \to \mathrm{BIN}, \quad G = (\mathrm{UN} \to \mathrm{UN}) \to \mathrm{UN},\\
\mathtt{add} &= \mathtt{fix}_F(\lambda f{:}\,\mathrm{BIN}\,.\lambda x{:}\mathtt{nat}.\lambda y{:}\mathtt{nat}.\mathtt{cond}_{\mathtt{nat}}((\mathtt{iszero}(y)))(x)(f(\mathtt{succ}(x))(\mathtt{pred}(y)))),\\
\mathtt{mult} &= \mathtt{fix}_F(\lambda f{:}\,\mathrm{BIN}\,.\lambda x{:}\mathtt{nat}.\lambda y{:}\mathtt{nat}.\mathtt{cond}_{\mathtt{nat}}((\mathtt{iszero}(y)))(y)(\mathtt{add}(fx(\mathtt{pred}(y)))x)),\\
\mathtt{factorial} &= \mathtt{fix}_G(\lambda f{:}\,\mathrm{UN}\,.\lambda n{:}\mathtt{nat}.\mathtt{cond}_{\mathtt{nat}}((\mathtt{iszero}(n)))(\mathit{succ}(0))(\mathtt{mult}n(f(\mathtt{pred}(n))))).
\end{aligned}$$

In fact, we have a very strong result for PCF[9]:

Theorem 2.36. *The computable functions are precisely the PCF-definable functions.*

Proof. See Plotkin (1977 p. 251). □

We have not been terribly formal about computability theory, but it is clear that PCF would suffice to provide a basis for functional languages without losing any of the computational power of the untyped λ-calculus: by Church-Turing thesis, we can write any algorithm in PCF. For a reference on computability, consult Sipser (2012). Henceforth, we will focus on $\lambda^{\to\times}$-theories as our object of study.

[9] This kind of result is colloquially known as *Turing completeness*.

3 CATEGORICAL SEMANTICS

We seek to *model* our $\lambda^{\to\times}$-theory categorically, by giving it a foundation in category theory. We will follow the approach of Crole (1993 Chapter 4); for details of the derivation of such semantics, see Crole (1993 pp. 163–168). Basic category theory, such as the definition of a category, functors, products, exponentials (see Appendix A), natural transformations, and adjunctions will be assumed; the standard reference text is Mac Lane (1998), but Awodey (2010) and Leinster (2016) are more suitable for those with a computer science background. Pierce (1991) provides an elementary introduction, and a short (albeit slightly outdated) survey of category theory applied to computer science, and Milewski (2017) provides an informal exposition suited to programmers who are non-mathematicians.

Definition 3.1 (Model of $\lambda^{\to\times}$ in a Cartesian closed category). Given a $\lambda^{\to\times}$-signature σ, we define a model $\mathbb{M}$ in a Cartesian closed category $\mathscr{C}$ to be a *structure* generated by the following data: objects

- for every ground type $G \in \mathsf{TV}$ of σ, an associated object $[\![G]\!]$;

and morphisms

- for every constant $c{:}A$, an associated morphism $[\![c]\!] : 1 \to [\![A]\!]$;
- for every n-ary function symbol, where $n > 0$, $f{:}A_1 \times \cdots \times A_n \to B$, an associated morphism $[\![f]\!] : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![B]\!]$;

such that

- the unit type unit is associated to the terminal object 1;
- for every other simple type $A \in \mathsf{ST}(\mathsf{TV})$, association is given inductively by

$$[\![A \to B]\!] \coloneqq [\![A]\!] \Rightarrow [\![B]\!], \tag{CatSem-$\to$}$$

$$[\![A \times B]\!] \coloneqq [\![A]\!] \times [\![B]\!]. \tag{CatSem-$\times$}$$

Furthermore, typing contexts are also modelled by objects[10]:

$$[\![\Gamma]\!] = \begin{cases} 1, & \text{if } \Gamma = [], \\ [\![A_1]\!] \times \cdots \times [\![A_n]\!], & \text{if } \Gamma = [x_1{:}A_1, \dots, x_n{:}A_n]. \end{cases} \tag{CatSem-Γ}$$

Finally, the typed terms of σ are modelled as follows

$$\text{var}\ \frac{}{[\![\Gamma + [x{:}A] + \Gamma' \vdash x : A]\!] \coloneqq \pi_i : [\![\Gamma]\!] \times [\![A]\!] \times [\![\Gamma']\!] \to [\![A]\!]} \quad x{:}A \text{ is the } i\text{th element of } \Gamma + [x{:}A] + \Gamma'$$

$$\text{unit}\ \frac{}{[\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!] \coloneqq\, ! : [\![\Gamma]\!] \to 1}$$

$$\text{const}\ \frac{}{[\![\Gamma \vdash c : A]\!] \coloneqq [\![\Gamma]\!] \xrightarrow{!} 1 \xrightarrow{[\![c]\!]} [\![A]\!]} \quad c \text{ is a constant of type } A$$

$$\text{func}\ \frac{[\![\Gamma \vdash t_1 : A_1]\!] = [\![t_1]\!] : [\![\Gamma]\!] \to [\![A_1]\!] \quad \dots \quad [\![\Gamma \vdash t_n : A_n]\!] = [\![t_n]\!] : [\![\Gamma]\!] \to [\![A_n]\!]}{[\![\Gamma \vdash f(t_1, \dots, t_n) : B]\!] \coloneqq [\![\Gamma]\!] \xrightarrow{\langle [\![t_1]\!], \dots, [\![t_n]\!] \rangle} [\![A_1]\!] \times \cdots \times [\![A_n]\!] \xrightarrow{[\![f]\!]} [\![B]\!]} \quad \begin{array}{c} f \text{ is an } n\text{-ary function} \\ \text{symbol of type} \\ A_1 \times \cdots \times A_n \to B \end{array}$$

$$\text{abs}\ \frac{[\![\Gamma + [x{:}A] \vdash t : B]\!] = s : [\![\Gamma]\!] \times [\![A]\!] \to [\![B]\!]}{[\![\Gamma \vdash (\lambda x{:}A.t) : A \to B]\!] \coloneqq \mathsf{curry}\,(s) : [\![\Gamma]\!] \to [\![A]\!] \Rightarrow [\![B]\!]}$$

$$\text{app}\ \frac{[\![\Gamma \vdash u : A \to B]\!] = [\![u]\!] : [\![\Gamma]\!] \to [\![A]\!] \Rightarrow [\![B]\!] \qquad [\![\Gamma \vdash v : A]\!] = [\![v]\!] : [\![\Gamma]\!] \to [\![A]\!]}{[\![\Gamma \vdash uv : B]\!] \coloneqq [\![\Gamma]\!] \xrightarrow{\langle [\![u]\!], [\![v]\!] \rangle} [\![A]\!] \Rightarrow [\![B]\!] \times [\![A]\!] \xrightarrow{\mathsf{ev}_{[\![A]\!],[\![B]\!]}} [\![B]\!]}$$

$$\text{pair}\ \frac{[\![\Gamma \vdash u : A]\!] = [\![u]\!] : [\![\Gamma]\!] \to [\![A]\!] \qquad [\![\Gamma \vdash v : B]\!] = [\![v]\!] : [\![\Gamma]\!] \to [\![B]\!]}{[\![\Gamma \vdash \langle u, v \rangle : A \times B]\!] \coloneqq \langle [\![u]\!], [\![v]\!] \rangle : [\![\Gamma]\!] \to [\![A]\!] \times [\![B]\!]}$$

[10] Categorical products conventionally associate to the left, so we write $X \times Y \times Z$ to mean $(X \times Y) \times Z$.

$$\mathsf{fst}\ \frac{[\![\Gamma \vdash t : A \times B]\!] = [\![t]\!] : [\![\Gamma]\!] \to [\![A]\!] \times [\![B]\!]}{[\![\Gamma \vdash \mathsf{fst}\,(t) : A]\!] \coloneqq [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![A]\!] \times [\![B]\!] \xrightarrow{\pi_1} [\![A]\!]}$$

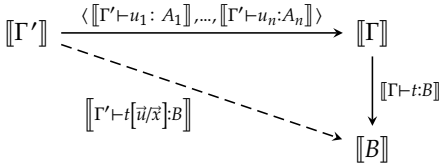
$$\mathsf{snd}\ \frac{[\![\Gamma \vdash t : A \times B]\!] = [\![t]\!] : [\![\Gamma]\!] \to [\![A]\!] \times [\![B]\!]}{[\![\Gamma \vdash \mathsf{snd}\,(t) : A]\!] \coloneqq [\![\Gamma]\!] \xrightarrow{[\![t]\!]} [\![A]\!] \times [\![B]\!] \xrightarrow{\pi_2} [\![B]\!]}$$

$\mathbb{M}$ *satisfies* an equation-in-context $\Gamma \vdash u = v : A$ if morphisms $[\![\Gamma \vdash u : A]\!]$ and $[\![\Gamma \vdash v : A]\!]$ are equal in $\mathscr{C}$, and furthermore $\mathbb{M}$ *models* $\mathcal{T} = (\sigma, \mathcal{A})$ if it satisfies all equations-in-context generated by $\mathcal{T}$. As a result of the Soundness Theorem (to be shown), this coincides exactly with $\mathbb{M}$ satisfying $\mathcal{A}$.

Lemma 3.2 (Semantics of simultaneous substitution). *Given a model* $\mathbb{M}$ *in a Cartesian closed category* $\mathscr{C}$*, such that* $\Gamma \vdash t : B$*, where* $\Gamma = [x_1{:}A_1, \dots, x_n{:}A_n]$ *is modelled by*

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

and there is $\Gamma' \vdash u_i : A_i$ *for* $i \in \{1, \dots, n\}$*, there exists a morphism* $\left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right]$ *such that the following diagram commutes:*



Proof. By induction on the derivation of $\Gamma \vdash t : B$, writing γ for $\langle [\![\Gamma' \vdash u_i : A_1]\!], \dots, [\![\Gamma' \vdash u_i : A_1]\!] \rangle$.

var case If $t = x_i$ for some $i \in \{1, \dots, n\}$ is a variable, then

$$\begin{aligned}
&\left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
=\ &\{\ \text{substitution}\ \} \\
&[\![\Gamma' \vdash u_i : A_i]\!] \\
=\ &\{\ \text{universal property of product}\ \} \\
&\pi_i \circ \gamma \\
=\ &\{\ \text{var}\ \} \\
&[\![\Gamma \vdash x_i : A_i]\!] \circ \gamma.
\end{aligned}$$

◁

unit case If t is $\langle\rangle$, then

$$\begin{aligned}
&\left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
=\ &\{\ \text{substitution}\ \} \\
&[\![\Gamma' \vdash \langle\rangle : \mathsf{unit}]\!] \\
=\ &\{\ \text{unit}\ \} \\
&!_{[\![\Gamma']\!]} \\
=\ &\{\ \text{universal property of terminal object}\ \} \\
&!_{[\![\Gamma]\!]} \circ \gamma \\
=\ &\{\ \text{unit}\ \}
\end{aligned}$$

$$[\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!] \circ \gamma.$$

◁

const case If t is a constant, then

$$\begin{aligned}
& \left[\!\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\!\right] \\
= \quad & \{ \text{ substitution } \} \\
& [\![\Gamma' \vdash t : B]\!] \\
= \quad & \{ \text{ const } \} \\
& [\![t]\!] \circ !_{[\![\Gamma']\!]} \\
= \quad & \{ \text{ universal property of terminal object } \} \\
& [\![t]\!] \circ !_{[\![\Gamma]\!]} \circ \gamma \\
= \quad & \{ \text{ const } \} \\
& [\![\Gamma \vdash t : B]\!] \circ \gamma.
\end{aligned}$$

◁

func case If t is an application of a n-ary function symbol s to v_i for $i \in \{1, \dots, n\}$, $s(v_1, \dots, v_n)$, such that there exist $[\![\Gamma \vdash v_i : C_i]\!] \colon [\![\Gamma]\!] \to [\![C_i]\!]$, and $[\![s]\!] \colon [\![C_1]\!] \times \cdots \times [\![C_n]\!] \to [\![B]\!]$, then

$$\begin{aligned}
& \left[\!\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\!\right] \\
= \quad & \{ \text{ substitution } \} \\
& \left[\!\!\left[\Gamma' \vdash s(v_1\left[\vec{u}/\vec{x}\right], \dots, v_n\left[\vec{u}/\vec{x}\right]) : B\right]\!\!\right] \\
= \quad & \{ \text{ func } \} \\
& [\![s]\!] \circ \langle \left[\!\!\left[\Gamma' \vdash v_1\left[\vec{u}/\vec{x}\right] : C_1\right]\!\!\right], \dots, \left[\!\!\left[\Gamma' \vdash v_n\left[\vec{u}/\vec{x}\right] : C_n\right]\!\!\right] \rangle \\
& \qquad\qquad \text{where } \left[\!\!\left[\Gamma' \vdash v_i\left[\vec{u}/\vec{x}\right] : C_i\right]\!\!\right] : [\![\Gamma']\!] \to [\![C_i]\!] \\
= \quad & \{ \text{ inductive hypothesis: } \left[\!\!\left[\Gamma' \vdash v_i\left[\vec{u}/\vec{x}\right] : C_i\right]\!\!\right] = [\![\Gamma \vdash v_i : C_i]\!] \circ \gamma \ \} \\
& [\![s]\!] \circ \langle [\![\Gamma \vdash v_1 : C_1]\!] \circ \gamma, \dots, [\![\Gamma \vdash v_n : C_n]\!] \circ \gamma \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& [\![s]\!] \circ \langle [\![\Gamma \vdash v_1 : C_1]\!], \dots, [\![\Gamma \vdash v_n : C_n]\!] \rangle \circ \gamma \\
= \quad & \{ \text{ func } \} \\
& [\![\Gamma \vdash s(v_1, \dots, v_n) : B]\!] \circ \gamma.
\end{aligned}$$

◁

abs case Firstly, consider the following lemma about currying.

Lemma 3.3. *In any Cartesian closed category, with morphisms $f \colon X \times Y \to Z$ and $g \colon W \to X$,*

$$\mathsf{curry}\left(f \circ g \times \mathrm{id}_Y\right) = \mathsf{curry}\left(f\right) \circ g.$$

See proof of Theorem A.4 for a proof.

If t is an abstraction $\lambda y{:}C.s$, such that $B = C \to D$ and there exist $[\![\Gamma + [x{:}C] \vdash s : B]\!] : [\![\Gamma]\!] \times [\![C]\!] \to [\![D]\!]$, then

$$
\begin{aligned}
& \left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
= \quad & \{ \text{ substitution } \} \\
& \left[\!\left[\Gamma' \vdash \lambda y{:}C.s\left[\vec{u}/\vec{x}\right] : C \to D\right]\!\right] \\
= \quad & \{ \text{ abs } \} \\
& \mathsf{curry}\left(\left[\!\left[\Gamma' + [y{:}C] \vdash s\left[\vec{u}/\vec{x}\right] : D\right]\!\right]\right) \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& \mathsf{curry}\left(\left[\!\left[\Gamma + [y{:}C] \vdash s : D\right]\!\right] \circ \gamma \times \mathrm{id}_{[\![C]\!]}\right) \\
= \quad & \{ \text{ Lemma 3.3 } \} \\
& \mathsf{curry}\left(\left[\!\left[\Gamma + [y{:}C] \vdash s : D\right]\!\right]\right) \circ \gamma \\
= \quad & \{ \text{ abs } \} \\
& [\![\Gamma \vdash \lambda y{:}C.s : C \to D]\!] \circ \gamma.
\end{aligned}
$$

◁

app case If t is an application sv, such that there exist $[\![\Gamma \vdash s : C \to B]\!] : [\![\Gamma]\!] \to [\![C]\!] \Rightarrow [\![B]\!]$ and $[\![\Gamma \vdash v : C]\!] : [\![\Gamma]\!] \to [\![C]\!]$, then

$$
\begin{aligned}
& \left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
= \quad & \{ \text{ substitution } \} \\
& \left[\!\left[\Gamma' \vdash s\left[\vec{u}/\vec{x}\right] v\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
= \quad & \{ \text{ app } \} \\
& \mathsf{ev}_{[\![C]\!],[\![B]\!]} \circ \langle \left[\!\left[\Gamma' \vdash s\left[\vec{u}/\vec{x}\right] : C \to B\right]\!\right], \left[\!\left[\Gamma' \vdash v\left[\vec{u}/\vec{x}\right] : C\right]\!\right] \rangle \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& \mathsf{ev}_{[\![C]\!],[\![B]\!]} \circ \langle [\![\Gamma \vdash s : C \to B]\!] \circ \gamma, [\![\Gamma \vdash v : C]\!] \circ \gamma \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& \mathsf{ev}_{[\![C]\!],[\![B]\!]} \circ \langle [\![\Gamma \vdash s : C \to B]\!], [\![\Gamma \vdash v : C]\!] \rangle \circ \gamma \\
= \quad & \{ \text{ app } \} \\
& [\![\Gamma \vdash sv : B]\!] \circ \gamma.
\end{aligned}
$$

◁

pair case If t is a pair $\langle p, q \rangle$, such that $B = P \times Q$ and there exist $\left[\!\left[\Gamma \vdash p : P\right]\!\right] : [\![\Gamma]\!] \to [\![P]\!]$ and $\left[\!\left[\Gamma \vdash q : Q\right]\!\right] : [\![\Gamma]\!] \to [\![Q]\!]$, then

$$
\begin{aligned}
& \left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
= \quad & \{ \text{ substitution } \}
\end{aligned}
$$

$$
\begin{aligned}
& \left[\!\left[\Gamma' \vdash \langle p\left[\vec{u}/\vec{x}\right], q\left[\vec{u}/\vec{x}\right]\rangle : P \times Q\right]\!\right] \\
= \quad & \{\ \text{pair}\ \} \\
& \langle \left[\!\left[\Gamma' \vdash p\left[\vec{u}/\vec{x}\right] : P\right]\!\right], \left[\!\left[\Gamma' \vdash q\left[\vec{u}/\vec{x}\right] : Q\right]\!\right] \rangle \\
= \quad & \{\ \text{inductive hypothesis}\ \} \\
& \langle [\![\Gamma \vdash p : P]\!] \circ \gamma, [\![\Gamma \vdash q : Q]\!] \circ \gamma \rangle \\
= \quad & \{\ \text{composition distributes over product}\ \} \\
& \langle [\![\Gamma \vdash p : P]\!], [\![\Gamma \vdash q : Q]\!] \rangle \circ \gamma \\
= \quad & \{\ \text{pair}\ \} \\
& [\![\Gamma \vdash \langle p, q\rangle : P \times Q]\!] \circ \gamma.
\end{aligned}
$$

◁

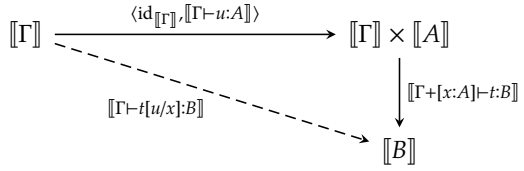
fst, snd case If t is the first element of a pair v, such that there exists $[\![\Gamma \vdash v : B \times C]\!] : [\![\Gamma]\!] \to [\![B]\!] \times [\![C]\!]$, then

$$
\begin{aligned}
& \left[\!\left[\Gamma' \vdash t\left[\vec{u}/\vec{x}\right] : B\right]\!\right] \\
= \quad & \{\ \text{substitution}\ \} \\
& \left[\!\left[\Gamma' \vdash \mathsf{fst}\left(v\left[\vec{u}/\vec{x}\right]\right) : B\right]\!\right] \\
= \quad & \{\ \text{fst}\ \} \\
& \pi_1 \circ \left[\!\left[\Gamma' \vdash v\left[\vec{u}/\vec{x}\right] : B \times C\right]\!\right] \\
= \quad & \{\ \text{inductive hypothesis}\ \} \\
& \pi_1 \circ [\![\Gamma \vdash v : B \times C]\!] \circ \gamma \\
= \quad & \{\ \text{fst}\ \} \\
& [\![\Gamma \vdash \mathsf{fst}\,(v) : B]\!] \circ \gamma.
\end{aligned}
$$

The case for **snd** is similar. ◁

□

Corollary 3.4 (Semantics of singular substitution). *Given a model* $\mathbb{M}$ *in a Cartesian closed category* $\mathscr{C}$*, such that* $\Gamma + [x{:}A] \vdash t : B$ *and* $\Gamma \vdash u : A$*, there exists a morphism* $[\![\Gamma \vdash t[u/x] : B]\!]$ *to make the following diagram commute:*



Proof. Follows as a special case of Lemma 3.2.

□

This lemma tells us that substitution (which is how β-reduction is defined) is interpreted by a composite of two morphisms in $\mathbb{M}$, and is fundamental to the β case when we come to prove soundness.

4 CATEGORICAL TYPE THEORY CORRESPONDENCE

The Curry-Howard-Lambek correspondence postulates a three-way isomorphism between type theory, proof theory, and category theory, which is informally referred to as *computational trinitarianism* (Harper 2011):

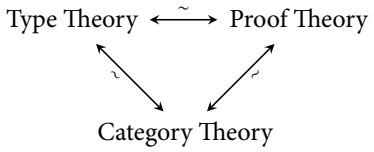


Fig. 2. The 'Holy Trinity'

The categorical type theory correspondence is the notion that category theory and type theory are intrinsically related; the bottom left edge, and the specific relationship explored by this thesis is the one between Cartesian closed categories and $\lambda^{\to\times}$-theories, in the sense that both are morally 'the same'. In the sequel, we explore more precisely what this means.

4.1 Soundness

The construction described in Definition 3.1 does form a category, although we would like to know that it faithfully reflects equations generated by a $\lambda^{\to\times}$-theory $\mathcal{T}$ — a *soundness* theorem expressing that the categorical interpretation of $\mathcal{T}$ suffices as a valid semantics for $\mathcal{T}$. For instance, given an equation-in-context $\Gamma \vdash u = v : A$, we would like their interpretations $[\![\Gamma \vdash u : A]\!]$ and $[\![\Gamma \vdash v : A]\!]$ to be 'the same'. As the interpretations are morphisms, the appropriate notion of 'sameness' is an identification of morphism equality in the category.

Theorem 4.1 (Soundness of categorical semantics). *Given a Cartesian closed category $\mathscr{C}$ and $\lambda^{\to\times}$-theory $\mathcal{T} = (\sigma, \mathcal{A})$, any structure of $\lambda^{\to\times}$-signature σ, $\mathbb{M}$, in $\mathscr{C}$* models *$\mathcal{T}$. That is to say, $\mathbb{M}$ satisfies all equations-in-context of $\mathcal{A}$.*

Equivalently, for terms-in-context $\Gamma \vdash u : A$ and $\Gamma \vdash v : A$,

$$\Gamma \vdash u = v : A \implies [\![\Gamma \vdash u : A]\!] = [\![\Gamma \vdash v : A]\!].$$

Proof. We proceed by induction on the theorems derivable in $\mathcal{T}$ (see Definition 2.29).

axiom case Trivial. ◁

refl, sym, trans cases Follows from $=$ being an equivalence relation on morphisms of $\mathscr{C}$. ◁

unit case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma \vdash t : \mathsf{unit}$, which is modelled by

$$[\![\Gamma \vdash t : \mathsf{unit}]\!] = !_{[\![\Gamma]\!]},$$

and so $\mathbb{M}$ satisfies $\Gamma \vdash t = \langle\rangle : \mathsf{unit}$ due to the universal property of terminal object. ◁

β case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma + [x{:}A] \vdash u : B$ and $\Gamma \vdash v : A$, modelled by

$$[\![\Gamma + [x{:}A] \vdash u : B]\!] : [\![\Gamma]\!] \times [\![A]\!] \to [\![B]\!],$$

and

$$[\![\Gamma \vdash v : A]\!] : [\![\Gamma]\!] \to [\![A]\!],$$

so we have

$$\begin{aligned} & [\![\Gamma \vdash (\lambda x{:}A.u)v : B]\!] \\ = \quad & \{\ \text{app, } [\![\Gamma \vdash v : A]\!] \text{ exists by inductive hypothesis } \} \end{aligned}$$

$$
\begin{aligned}
& \mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ \langle [\![\Gamma \vdash (\lambda x{:}A.u) : A \to B]\!], [\![\Gamma \vdash v : A]\!] \rangle \\
= \quad & \{ \text{ abs, } [\![\Gamma + [x{:}A] \vdash u : B]\!] \text{ exists by inductive hypothesis } \} \\
& \mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ \langle \mathsf{curry}\left([\![\Gamma + [x{:}A] \vdash u : B]\!]\right), [\![\Gamma \vdash v : A]\!] \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& \mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ \mathsf{curry}\left([\![\Gamma + [x{:}A] \vdash u : B]\!]\right) \times \mathrm{id}_{[\![A]\!]} \circ \langle \mathrm{id}_{[\![\Gamma]\!]}, [\![\Gamma \vdash v : A]\!] \rangle \\
= \quad & \{ \text{ uncurry } \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left([\![\Gamma + [x{:}A] \vdash u : B]\!]\right)\right) \circ \langle \mathrm{id}_{[\![\Gamma]\!]}, [\![\Gamma \vdash v : A]\!] \rangle \\
= \quad & \{ \text{ universal property of exponential } \} \\
& [\![\Gamma + [x{:}A] \vdash u : B]\!] \circ \langle \mathrm{id}_{[\![\Gamma]\!]}, [\![\Gamma \vdash v : A]\!] \rangle \\
= \quad & \{ \text{ Corollary 3.4 } \} \\
& [\![\Gamma \vdash u[v/x] : B]\!].
\end{aligned}
$$

◁

η **case** By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma \vdash t : A \to B$, modelled by

$$[\![\Gamma \vdash t : A \to B]\!] \colon [\![\Gamma]\!] \to [\![A]\!] \Rightarrow [\![B]\!],$$

so we have

$$
\begin{aligned}
& [\![\Gamma \vdash (\lambda x{:}A.tx) : A \to B]\!] \\
= \quad & \{ \text{ abs } \} \\
& \mathsf{curry}\left([\![\Gamma + [x{:}A] \vdash tx : B]\!]\right) \\
= \quad & \{ \text{ app } \} \\
& \mathsf{curry}\left(\mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ \langle [\![\Gamma + [x{:}A] \vdash t : A \to B]\!], [\![\Gamma + [x{:}A] \vdash x : A]\!] \rangle\right) \\
= \quad & \{ \text{ var, } [\![\Gamma \vdash t : A \to B]\!] \text{ exists by inductive hypothesis } \} \\
& \mathsf{curry}\left(\mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ \langle [\![\Gamma \vdash t : A \to B]\!] \circ \pi_1, \pi_2 \rangle\right) \\
= \quad & \{ \; - \times - \text{ for morphisms } \} \\
& \mathsf{curry}\left(\mathsf{ev}_{[\![A]\!],[\![B]\!]} \circ [\![\Gamma \vdash t : A \to B]\!] \times \mathrm{id}_{[\![A]\!]}\right) \\
= \quad & \{ \text{ uncurry } \} \\
& \mathsf{curry}\left(\mathsf{uncurry}\left([\![\Gamma \vdash t : A \to B]\!]\right)\right) \\
= \quad & \{ \text{ universal property of exponential } \} \\
& [\![\Gamma \vdash t : A \to B]\!].
\end{aligned}
$$

◁

abs case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma + [x{:}A] \vdash u = v : B$, so we have

$$
\begin{aligned}
& [\![\Gamma \vdash \lambda x{:}A.u : A \to B]\!] \\
= \quad & \{ \text{ abs } \} \\
& \mathsf{curry}\left([\![\Gamma + [x{:}A] \vdash u : B]\!]\right) \\
= \quad & \{ \text{ inductive hypothesis } \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{curry}\left(\llbracket \Gamma + [x{:}A] \vdash v : B \rrbracket\right) \\
= \quad & \{\ \text{abs}\ \} \\
& \llbracket \Gamma \vdash \lambda x{:}A.v : A \to B \rrbracket.
\end{aligned}
$$

◁

app case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma \vdash s = t : A \to B$ and $\Gamma \vdash u = v : A$, so we have

$$
\begin{aligned}
& \llbracket \Gamma \vdash su : B \rrbracket \\
= \quad & \{\ \text{app}\ \} \\
& \mathsf{ev}_{\llbracket A \rrbracket, \llbracket B \rrbracket} \circ \langle \llbracket \Gamma \vdash s : A \to B \rrbracket, \llbracket \Gamma \vdash u : A \rrbracket \rangle \\
= \quad & \{\ \text{inductive hypothesis}\ \} \\
& \mathsf{ev}_{\llbracket A \rrbracket, \llbracket B \rrbracket} \circ \langle \llbracket \Gamma \vdash t : A \to B \rrbracket, \llbracket \Gamma \vdash v : A \rrbracket \rangle \\
= \quad & \{\ \text{app}\ \} \\
& \llbracket \Gamma \vdash tv : B \rrbracket.
\end{aligned}
$$

◁

pair case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma \vdash t : A \times B$, so we have

$$
\begin{aligned}
& \llbracket \Gamma \vdash \langle \mathsf{fst}\,(t)\,, \mathsf{snd}\,(t) \rangle : A \times B \rrbracket \\
= \quad & \{\ \text{pair}\ \} \\
& \langle \llbracket \Gamma \vdash \mathsf{fst}\,(t) : A \rrbracket, \llbracket \Gamma \vdash \mathsf{snd}\,(t) : B \rrbracket \rangle \\
= \quad & \{\ \mathsf{fst}\ \text{and}\ \mathsf{snd}\ \} \\
& \langle \pi_1 \circ \llbracket \Gamma \vdash t : A \times B \rrbracket, \pi_2 \circ \llbracket \Gamma \vdash t : A \times B \rrbracket \rangle \\
= \quad & \{\ \text{universal property of product}\ \} \\
& \llbracket \Gamma \vdash t : A \times B \rrbracket.
\end{aligned}
$$

◁

fst, snd case By the inductive hypothesis, $\mathbb{M}$ satisfies $\Gamma \vdash u : A$ and $\Gamma \vdash v : B$, so we have

$$
\begin{aligned}
& \llbracket \Gamma \vdash \mathsf{fst}\,(\langle u, v \rangle) : A \rrbracket \\
= \quad & \{\ \mathsf{fst}\ \} \\
& \pi_1 \circ \llbracket \Gamma \vdash \langle u, v \rangle : A \times B \rrbracket \\
= \quad & \{\ \text{pair}\ \} \\
& \pi_1 \circ \langle \llbracket \Gamma \vdash u : A \rrbracket, \llbracket \Gamma \vdash v : B \rrbracket \rangle \\
= \quad & \{\ \text{universal property of product}\ \} \\
& \llbracket \Gamma \vdash u : A \rrbracket.
\end{aligned}
$$

The case for snd is similar. ◁

This covers all the cases for derivable theorems in $\mathcal{T}$, and so $\mathbb{M}$ satisfies all equations generated by $\mathcal{T}$. □

Corollary 4.2. *Every Cartesian closed category $\mathscr{C}$ gives rise to a $\lambda^{\to\times}$-theory* **Lan** $(\mathscr{C})$. *Such a theory is called the* ***internal language*** *of $\mathscr{C}$.*

Proof. We will define **Lan** $(\mathscr{C}) = (\sigma(\mathscr{C}), \mathcal{A}(\mathscr{C}))$ by its signature and then its axioms. Its signature has ground types the objects of $\mathscr{C}$, and function symbols for the morphisms of $\mathscr{C}$ (where constants correspond to global elements of $\mathscr{C}$). Furthermore, for each object X of $\mathscr{C}$, we define two function symbols

$$\mathsf{model}_X \colon X \to [\![X]\!],$$
$$\mathsf{unmodel}_X \colon [\![X]\!] \to X,$$

where $[\![X]\!]$ is the interpretation of X in a structure of $\sigma(\mathscr{C})$ in $\mathscr{C}$. A canonical choice for such a structure assigns

- $[\![G]\!] = G$ for all ground types G of $\sigma(\mathscr{C})$, ensuring that by definition of categorical semantics that $[\![X]\!] = X$ for all types generated by the signature;
- $[\![f]\!] = f$ for all function symbols and constants;
- $[\![\mathsf{model}_X]\!] = [\![\mathsf{unmodel}_X]\!] = \mathrm{id}_X$.

Now let $\mathcal{A}(\mathscr{C})$ be the set of equations-in-context satisfied by $\sigma(\mathscr{C})$, thus such a structure is a model of **Lan** $(\mathscr{C})$.

□

The internal language of a category can be used to reason about it; indeed, a big part of the categorical type theory correspondence is that any derivable theorem from the perspective of $\lambda^{\to\times}$-theory has a meaningful interpretation in a Cartesian closed category, and vice versa.

4.2 Completeness

As all Cartesian closed categories model any given $\lambda^{\to\times}$-theory $\mathcal{T}$, a natural follow up to the Soundness Theorem is to consider how all models of $\mathcal{T}$ in a specific Cartesian closed category relate. The answer is that they form a category of models, with model homomorphisms as morphisms, which we explore in due course.

Furthermore, given a $\lambda^{\to\times}$-theory $\mathcal{T}$, we would like to construct a Cartesian closed category which contains a model of it — better still, the 'smallest' such category, in the sense of a Cartesian closed category *freely* generated from $\mathcal{T}$, formalising the notion of 'no unnecessary information'. Such a category is called the *classifying category* of $\mathcal{T}$, denoted **Syn** $(\mathcal{T})$, and accompanying this construction is a *completeness* theorem ensuring the adequacy of **Syn** $(\mathcal{T})$ in modelling $\mathcal{T}$.

Firstly, we develop the notion of a Cartesian closed functor.

Definition 4.3 (Product-preserving functor). A functor $F \colon \mathscr{C} \to \mathscr{D}$ preserves finite products if and only if for each finite product $\prod_i^n X_i$ in $\mathscr{C}$ with projection maps $\pi_i \colon \prod_i^n X_i \to X_i$, there is an isomorphism in $\mathscr{D}$

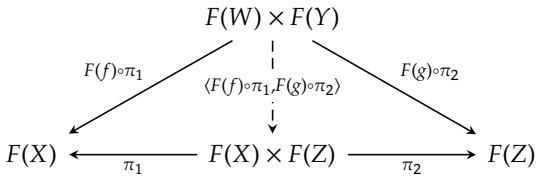
$$\Phi_{\prod_i^n X_i} \coloneqq \langle F(\pi_1), \dots, F(\pi_n) \rangle \colon F\left(\prod_i^n X_i\right) \xrightarrow{\sim} \prod_i^n F(X_i). \tag{PP-Functor}$$

In the case where $n = 0$, we say that F preserves the terminal object, with $\Phi \colon F(1_{\mathscr{C}}) \xrightarrow{\sim} 1_{\mathscr{D}}$. F is *strict* if each isomorphism is an identity.

Lemma 4.4. *Given morphisms $f \colon W \to X$, $g \colon Y \to Z$, and a product-preserving functor $F \colon \mathscr{C} \to \mathscr{D}$, the following diagram commutes:*

$$\begin{array}{ccc} F(W \times Y) & \xrightarrow[\Phi_{W\times Y}]{\sim} & F(W) \times F(Y) \\ \Big\downarrow{\scriptstyle F(f\times g)} & & \Big\downarrow{\scriptstyle F(f)\times F(g)} \\ F(X \times Z) & \xrightarrow[\Phi_{X\times Z}]{\sim} & F(X) \times F(Z) \end{array}$$

Proof. Consider the universal property of product in $\mathscr{D}$:



So the morphism $\langle F(f) \circ \pi_1, F(g) \circ \pi_2 \rangle = F(f) \times F(g)$ is unique in $\mathscr{D}$, and hence equal to $\Phi_{X\times Z} \circ F(f \times g) \circ \Phi^{-1}_{W\times Y}$. Composition with inverses yields $F(f \times g) = \Phi^{-1}_{X\times Z} \circ F(f) \times F(g) \circ \Phi_{W\times Y}$.

□

Definition 4.5 (Cartesian closed functor). A functor $F\colon \mathscr{C} \to \mathscr{D}$ is Cartesian closed if and only if it preserves finite products and exponentials; it is a product-preserving functor such that additionally there is an isomorphism in $\mathscr{D}$

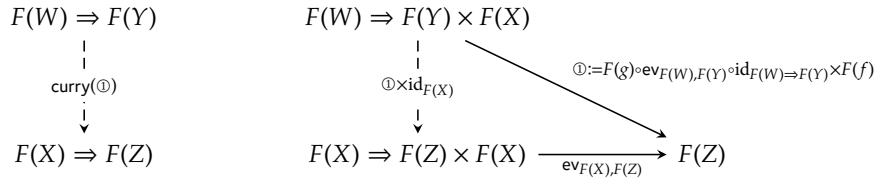
$$\Psi_{A\Rightarrow B} \coloneqq \mathsf{curry}\left(F(\mathsf{ev}_{A,B}) \circ \Phi^{-1}_{A\Rightarrow B\times A}\right) : F(A \Rightarrow B) \xrightarrow{\sim} F(A) \Rightarrow F(B). \tag{CC-Functor}$$

F is strict if each isomorphism is an identity.

Lemma 4.6. *Given morphisms $f\colon W \to X$, $g\colon Y \to Z$, and a Cartesian closed functor $F\colon \mathscr{C} \to \mathscr{D}$, the following diagram commutes:*

$$\begin{array}{ccc} F(W \Rightarrow Y) & \xrightarrow[\Psi_{W\Rightarrow Y}]{\sim} & F(W) \Rightarrow F(Y) \\ \Big\downarrow{\scriptstyle F(f\Rightarrow g)} & & \Big\downarrow{\scriptstyle F(f)\Rightarrow F(g)} \\ F(X \Rightarrow Z) & \xrightarrow[\Psi_{X\Rightarrow Z}]{\sim} & F(X) \Rightarrow F(Z) \end{array}$$

Proof. Consider the universal property of exponential in $\mathscr{D}$:



So the morphism $\mathsf{curry}\left(F(g) \circ \mathsf{ev}_{F(W),F(Y)} \circ \mathrm{id}_{F(W)\Rightarrow F(Y)} \times F(f)\right) = F(f) \Rightarrow F(g)$ is unique in $\mathscr{D}$, and hence equal to $\Psi_{X\Rightarrow Z} \circ F(f \Rightarrow g) \circ \Psi^{-1}_{W\Rightarrow Y}$. Composition with inverses yields $F(f \Rightarrow g) = \Psi^{-1}_{X\Rightarrow Z} \circ F(f) \Rightarrow F(g) \circ \Psi_{W\Rightarrow Y}$.

□

Definition 4.7 (Naturally isomorphic Cartesian closed functor category). The category of *naturally isomorphic Cartesian closed functors* between two Cartesian closed categories $\mathscr{C}$ and $\mathscr{D}$ is the category with Cartesian closed functors as its objects and natural isomorphisms as its morphisms, written

$$\mathbf{CCCat}_{\simeq}(\mathscr{C}, \mathscr{D}).$$

This is a subcategory of the functor category $[\mathscr{C}, \mathscr{D}]$, obtained by removing natural transformations which are not isomorphisms.

More machinery is required to develop the completeness theorem.

4.2.1 Syntactic category. Secondly, we describe the construction of a 'minimal' category from a $\lambda^{\to\times}$-theory.

Definition 4.8 (Syntactic category). Let $\mathcal{T}$ be a $\lambda^{\to\times}$-theory, and introduce

$$\ulcorner t \urcorner^A :\equiv \vdash t : A$$

as notation for the closed term of t of type A. Then, we construct a *syntactic category* $\mathbf{Syn}(\mathcal{T})$ from only the syntax of $\mathcal{T}$ as follows:

- objects are given by the simple types of $\mathcal{T}$;
- morphisms are given by equivalence classes (under $\mathcal{T}$[11]) of closed terms of $\mathcal{T}$: $[\ulcorner t \urcorner^A]_{\mathcal{T}}$;
  - necessarily, for each constant $c{:}A$ and n-ary function symbol $f{:}A_1 \times \cdots \times A_n \to B$,

    $$[\ulcorner c \urcorner^A]_{\mathcal{T}} \quad \text{and} \quad [\ulcorner f \urcorner^{A_1 \times \cdots \times A_n \to B}]_{\mathcal{T}}$$

    exist;
- identity morphisms given by $\mathrm{id}_A :\equiv [\ulcorner \lambda x{:}A.x \urcorner^{A \to A}]_{\mathcal{T}}$;
- composition between $f = [\ulcorner u \urcorner^{A \to B}]_{\mathcal{T}}$ and $g = [\ulcorner v \urcorner^{B \to C}]_{\mathcal{T}}$ given by

  $$g \circ f :\equiv [\ulcorner \lambda x{:}A.g(fx) \urcorner^{A \to C}]_{\mathcal{T}}; \tag{Syn-$\circ$}$$

- a terminal object $1 :\equiv \mathsf{unit}$;
- products given by $A \times B :\equiv A \times B$[12] with projections $\pi_1 :\equiv [\ulcorner \mathsf{fst} \urcorner^{A \times B \to A}]_{\mathcal{T}}$ and $\pi_2 :\equiv [\ulcorner \mathsf{snd} \urcorner^{A \times B \to B}]_{\mathcal{T}}$;
- exponentials given by $A \Rightarrow B :\equiv A \to B$ with $\mathsf{ev}_{A,B} :\equiv [\ulcorner \lambda x{:}(A \to B) \times A.\, \mathsf{fst}\,(x)\, \mathsf{snd}\,(x) \urcorner^B]_{\mathcal{T}}$.

Proposition 4.9. $\mathbf{Syn}(\mathcal{T})$ *forms a Cartesian closed category.*

Proof. First, we must show that it forms a category: composition is associative, and that identities are indeed identities.

Take $f = [\ulcorner t \urcorner^{A \to B}]_{\mathcal{T}}$, $g = [\ulcorner u \urcorner^{B \to C}]_{\mathcal{T}}$, and $h = [\ulcorner v \urcorner^{C \to D}]_{\mathcal{T}}$; then

$$\begin{aligned}
& h \circ (g \circ f) \\
= \quad & \{\ \text{(Syn-}\circ\text{)}\ \} \\
& h \circ [\ulcorner \lambda x{:}A.u(tx) \urcorner^{A \to C}]_{\mathcal{T}} \\
= \quad & \{\ \text{(Syn-}\circ\text{)}\ \} \\
& [\ulcorner \lambda y{:}A.v(\lambda x{:}A.u(tx)y) \urcorner^{A \to D}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \}
\end{aligned}$$

[11] Such an equivalence is given by $\beta\eta$-equality plus equalities resulting from $\mathcal{A}$.

[12] The syntax $- \times -$ is overloaded here, meaning categorical product on the left and product type on the right.

$$
\begin{aligned}
& [\ulcorner \lambda y{:}A.v(u(ty))\urcorner^{A\to D}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.\lambda z{:}B.v(uz)(ty)\urcorner^{A\to D}]_{\mathcal{T}} \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& [\ulcorner \lambda z{:}B.v(uz)\urcorner^{B\to D}]_{\mathcal{T}} \circ f \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& (h \circ g) \circ f.
\end{aligned}
$$

It is also trivial to see that, for any morphism $f = [\ulcorner t\urcorner^{A\to B}]_{\mathcal{T}}$,

$$
\begin{aligned}
& \mathrm{id}_B \circ f \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& [\ulcorner \lambda x{:}A.(\lambda y{:}B.y)(tx)\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda x{:}A.tx\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& = f \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda x{:}A.t((\lambda z{:}A.z)x)\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& f \circ \mathrm{id}_A.
\end{aligned}
$$

Now, it remains to be shown that unit is indeed the terminal object, and that there exist unique (up to unique isomorphism) products and exponentials.

Consider any type A, and construct the term $!_A \coloneqq [\ulcorner \lambda x{:}A.\langle\rangle\urcorner^{A\to\mathsf{unit}}]_{\mathcal{T}}$, which is the constant function returning $\langle\rangle$; as we made no assumptions on A other than it is a type, such a morphism exists for all objects in $\mathbf{Syn}(\mathcal{T})$. Now, for uniqueness, it suffices to show that for any $f\colon A \to B = [\ulcorner t\urcorner^{A\to B}]_{\mathcal{T}}$, $!_B \circ f = !_A$:

$$
\begin{aligned}
& !_B \circ f \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& [\ulcorner \lambda y{:}A.(x{:}B.\langle\rangle)(ty)\urcorner^{A\to\mathsf{unit}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.\langle\rangle\urcorner^{A\to\mathsf{unit}}]_{\mathcal{T}} \\
= \quad & \{\ \text{definition of } !_A\ \} \\
& !_A.
\end{aligned}
$$

For products, we need to show that for any two morphisms $f\colon A \to B = [\ulcorner u\urcorner^{A\to B}]_{\mathcal{T}}$ and $g\colon A \to C = [\ulcorner v\urcorner^{A\to B}]_{\mathcal{T}}$, there is a unique morphism $\langle f, g\rangle\colon A \to B \times C$ through which f and g factor; i.e. $f = \pi_1 \circ \langle f, g\rangle$ and $g = \pi_2 \circ \langle f, g\rangle$. Construct

$$
\langle f, g\rangle \coloneqq [\ulcorner \lambda x{:}A.\langle ux, vx\rangle\urcorner^{A\to B\times C}]_{\mathcal{T}}, \qquad (\text{Syn-}\langle -, -\rangle)
$$

and then

$$\begin{aligned}
& \pi_1 \circ \langle f, g \rangle \\
= \quad & \{\ \text{(Syn-}\langle -,-\rangle\text{) and (Syn-}\circ\text{)}\ \} \\
& [\ulcorner \lambda y{:}A.\, \mathsf{fst}\left(((\lambda x{:}A.\langle ux, vx\rangle)y)\right)\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.\, \mathsf{fst}\left(\langle uy, vy\rangle\right)\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.uy\urcorner^{A\to B}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& f,
\end{aligned}$$

and similarly for g. For the uniqueness, it is sufficient to show that, given $f\colon A \to B \times C = [\ulcorner t\urcorner^{A\to B\times C}]_{\mathcal{T}}$, $f = \langle \pi_1 \circ f, \pi_2 \circ f\rangle$:

$$\begin{aligned}
& \langle \pi_1 \circ f, \pi_2 \circ f\rangle \\
= \quad & \{\ \text{(Syn-}\circ\text{)}\ \} \\
& \langle [\ulcorner \lambda x{:}A.\, \mathsf{fst}\,(tx)\urcorner^{A\to B}]_{\mathcal{T}}, [\ulcorner \lambda x{:}A.\, \mathsf{snd}\,(tx)\urcorner^{A\to C}]_{\mathcal{T}}\rangle \\
= \quad & \{\ \text{(Syn-}\langle -,-\rangle\text{)}\ \} \\
& [\ulcorner \lambda y{:}A.\langle(\lambda x{:}A.\, \mathsf{fst}\,(tx))y, (\lambda x{:}A.\, \mathsf{snd}\,(tx))y\rangle\urcorner^{A\to B\times C}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.\langle \mathsf{fst}\left(ty\right), \mathsf{snd}\left(ty\right)\rangle\urcorner^{A\to B\times C}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& [\ulcorner \lambda y{:}A.ty\urcorner^{A\to B\times C}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& f.
\end{aligned}$$

Finally, for the exponential, we must demonstrate, for every $f\colon A \times B \to C = [\ulcorner t\urcorner^{A\times B\to C}]_{\mathcal{T}}$, the existence of a morphism $\mathsf{curry}\left(f\right) : A \to B \Rightarrow C$ such that $f = \mathsf{ev}_{B,C} \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_B$. Construct

$$\mathsf{curry}\left(f\right) \coloneqq [\ulcorner \lambda x{:}A.\lambda y{:}B.t\langle x, y\rangle\urcorner^{A\to(B\to C)}]_{\mathcal{T}}, \tag{Syn-curry}$$

and observe that

$$\begin{aligned}
& \mathsf{ev}_{B,C} \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_B \\
= \quad & \{\ \text{(Syn-curry)}\ \} \\
& \mathsf{ev}_{B,C} \circ [\ulcorner \lambda x{:}A.\lambda y{:}B.t\langle x, y\rangle\urcorner^{A\to(B\to C)}]_{\mathcal{T}} \times \mathrm{id}_B \\
= \quad & \{\ \times\ \} \\
& \mathsf{ev}_{B,C} \circ \langle [\ulcorner \lambda x{:}A.\lambda y{:}B.t\langle x, y\rangle\urcorner^{A\to(B\to C)}]_{\mathcal{T}} \circ \pi_1, \mathrm{id}_B \circ \pi_2\rangle \\
= \quad & \{\ \text{(Syn-}\circ\text{)}\ \}
\end{aligned}$$

$$
\begin{aligned}
& \mathsf{ev}_{B,C} \circ \langle [\ulcorner \lambda z{:}A \times B.(\lambda x{:}A.\lambda y{:}B.t\langle x, y\rangle)(\mathsf{fst}\,(z))^{\urcorner^{A\times B\to(B\to C)}}]_{\mathcal{T}}, [\ulcorner \lambda x{:}A \times B.\,\mathsf{snd}\,(x)^{\urcorner^{A\times B\to B}}]_{\mathcal{T}}\rangle \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& \mathsf{ev}_{B,C} \circ \langle [\ulcorner \lambda z{:}A \times B.\lambda y{:}B.t\langle \mathsf{fst}\,(z)\,, y\rangle^{\urcorner^{A\times B\to(B\to C)}}]_{\mathcal{T}}, [\ulcorner \lambda x{:}A \times B.\,\mathsf{snd}\,(x)^{\urcorner^{A\times B\to B}}]_{\mathcal{T}}\rangle \\
= \quad & \{\ (\text{Syn-}\langle -,-\rangle)\ \} \\
& \mathsf{ev}_{B,C} \circ [\ulcorner \lambda w{:}A \times B.\langle(\lambda z{:}A \times B.\lambda y{:}B.t\langle \mathsf{fst}\,(z)\,, y\rangle)w, (\lambda x{:}A \times B.\,\mathsf{snd}\,(x))w\rangle^{\urcorner^{A\times B\to(B\to C)\times B}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& \mathsf{ev}_{B,C} \circ [\ulcorner \lambda w{:}A \times B.\langle\lambda y{:}B.t\langle \mathsf{fst}\,(w)\,, y\rangle, \mathsf{snd}\,(w)\rangle^{\urcorner^{A\times B\to(B\to C)\times B}}]_{\mathcal{T}} \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& [\ulcorner \lambda p{:}A \times B.(\lambda q{:}(B \to C) \times B.\,\mathsf{fst}\left(q\right)\mathsf{snd}\left(q\right))(\lambda w{:}A \times B.\langle\lambda y{:}B.t\langle \mathsf{fst}\,(w)\,, y\rangle, (\mathsf{snd}\,(w))\rangle p)^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A \times B.(\lambda q{:}(B \to C) \times B.\,\mathsf{fst}\left(q\right)\mathsf{snd}\left(q\right))\langle\lambda y{:}B.t\langle \mathsf{fst}\left(p\right), y\rangle, \mathsf{snd}\left(p\right)\rangle^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A \times B.\mathsf{fst}\left(\langle\lambda y{:}B.t\langle \mathsf{fst}\left(p\right), y\rangle, \mathsf{snd}\left(p\right)\rangle\right)\mathsf{snd}\left(\langle\lambda y{:}B.t\langle \mathsf{fst}\left(p\right), y\rangle, \mathsf{snd}\left(p\right)\rangle\right)^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A \times B.(\lambda y{:}B.t\langle \mathsf{fst}\left(p\right), y\rangle)\mathsf{snd}\left(p\right)^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A \times B.t\langle \mathsf{fst}\left(p\right), \mathsf{snd}\left(p\right)\rangle^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A \times B.tp^{\urcorner^{A\times B\to C}}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& f.
\end{aligned}
$$

For uniqueness, we must show that given an $f\colon A \to B \Rightarrow C = [\ulcorner t^{\urcorner^{A\to(B\to C)}}]_{\mathcal{T}}$, $f = \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ f \times \mathrm{id}_B\right)$:

$$
\begin{aligned}
& \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ f \times \mathrm{id}_B\right) \\
= \quad & \{\ \times\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ \langle f \circ \pi_1, \mathrm{id}_B \circ \pi_2\rangle\right) \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ \langle [\ulcorner \lambda x{:}A \times B.t\ \mathsf{fst}\,(x)^{\urcorner^{A\times B\to(B\to C)}}]_{\mathcal{T}}, [\ulcorner \lambda x{:}A \times B.\,\mathsf{snd}\,(x)^{\urcorner^{A\times B\to B}}]_{\mathcal{T}}\rangle\right) \\
= \quad & \{\ (\text{Syn-}\langle -,-\rangle)\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ [\ulcorner \lambda y{:}A \times B.\langle(\lambda x{:}A \times B.t\ \mathsf{fst}\,(x))y, (\lambda x{:}A \times B.\,\mathsf{snd}\,(x))y\rangle^{\urcorner^{A\times B\to(B\to C)\times B}}]_{\mathcal{T}}\right) \\
= \quad & \{\ \beta\text{-equivalence}\ \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{curry}\left(\mathsf{ev}_{A,B} \circ [\ulcorner \lambda y{:}A \times B.\langle t\ \mathsf{fst}\left(y\right), \mathsf{snd}\left(y\right)\rangle \urcorner^{A\times B\to(B\to C)\times B}]_{\mathcal{T}}\right) \\
= \quad & \{\ (\text{Syn-}\circ)\ \} \\
& \mathsf{curry}\left([\ulcorner \lambda w{:}A \times B.(\lambda z{:}A \to B \times A.\ \mathsf{fst}\,(z)\ \mathsf{snd}\,(z))((\lambda y{:}A \times B.\langle t\ \mathsf{fst}\left(y\right), \mathsf{snd}\left(y\right)\rangle)w) \urcorner^{A\times B\to C}]_{\mathcal{T}}\right) \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& \mathsf{curry}\left([\ulcorner \lambda w{:}A \times B.(\lambda z{:}A \to B \times A.\ \mathsf{fst}\,(z)\ \mathsf{snd}\,(z))\langle t\ \mathsf{fst}\,(w), \mathsf{snd}\,(w)\rangle \urcorner^{A\times B\to C}]_{\mathcal{T}}\right) \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& \mathsf{curry}\left([\ulcorner \lambda w{:}A \times B.\ \mathsf{fst}\left(\langle t\ \mathsf{fst}\,(w), \mathsf{snd}\,(w)\rangle\right)\ \mathsf{snd}\left(\langle t\ \mathsf{fst}\,(w), \mathsf{snd}\,(w)\rangle\right) \urcorner^{A\times B\to C}]_{\mathcal{T}}\right) \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& \mathsf{curry}\left([\ulcorner \lambda w{:}A \times B.(t\ \mathsf{fst}\,(w))\ \mathsf{snd}\,(w) \urcorner^{A\times B\to C}]_{\mathcal{T}}\right) \\
= \quad & \{\ (\text{Syn-curry})\ \} \\
& [\ulcorner \lambda p{:}A.\lambda q{:}B.(\lambda w{:}A \times B.(t\ \mathsf{fst}\,(w))\ \mathsf{snd}\,(w))\langle p, q\rangle \urcorner^{A\to(B\to C)}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A.\lambda q{:}B.(t\ \mathsf{fst}\left(\langle p, q\rangle\right))\ \mathsf{snd}\left(\langle p, q\rangle\right) \urcorner^{A\to(B\to C)}]_{\mathcal{T}} \\
= \quad & \{\ \beta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A.\lambda q{:}B.(tp)q \urcorner^{A\to(B\to C)}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& [\ulcorner \lambda p{:}A.tp \urcorner^{A\to(B\to C)}]_{\mathcal{T}} \\
= \quad & \{\ \eta\text{-equivalence}\ \} \\
& f.
\end{aligned}
$$

Therefore, **Syn** $(\mathcal{T})$ is a Cartesian closed category.

□

Syn $(\mathcal{T})$ is the free category generated by the syntax of $\mathcal{T}$ quotiented by $\mathcal{T}$-equivalence (which is the closure of $\mathcal{T}$ under $\beta\eta$-equivalence).

4.2.2 Category of models. Thirdly, we examine the structure between different models of any given $\lambda^{\to\times}$-theory confined to a specific Cartesian closed category.

Definition 4.10 (Model homomorphism). A *model homomorphism* is a structure-preserving map between models of a $\lambda^{\to\times}$-theory in a category. If $\mathbb{M}$ models $\mathcal{T}$ in $\mathscr{C}$, then we define a model homomorphism

$$h\colon \mathbb{M} \to \mathbb{N}$$

by a collection of morphisms which map interpretations of a type of $\mathcal{T}$ in $\mathbb{M}$ (objects) to interpretations of that type in $\mathbb{N}$, which commute with the interpretations of function symbols, product types, and function types.

Each map is called a component of h, and h is specified component-wise by

$$h_A\colon [\![A]\!]_{\mathbb{M}} \to [\![A]\!]_{\mathbb{N}}$$

for each type $A \in \mathsf{ST}\,(\mathsf{TV})$ of $\mathcal{T}$.

The commutativity conditions are expressed as follows: for each constant symbol $c{:}A$, the following diagram commutes

$$\begin{array}{ccc} [\![\mathsf{unit}]\!]_{\mathbb{M}} = [\![\mathsf{unit}]\!]_{\mathbb{N}} & \xrightarrow{[\![c]\!]_{\mathbb{M}}} & [\![A]\!]_{\mathbb{M}} \\ & \searrow^{[\![c]\!]_{\mathbb{N}}} & \downarrow{\scriptstyle h_A} \\ & & [\![A]\!]_{\mathbb{N}} \end{array} \tag{MHom-Const}$$

For an n-ary function symbol $f{:}A_1 \times \cdots \times A_n \to B$, the following diagram commutes

$$\begin{array}{ccc} [\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}} & \xrightarrow{[\![f]\!]_{\mathbb{M}}} & [\![B]\!]_{\mathbb{M}} \\ \downarrow{\scriptstyle h_{A_1} \times \cdots \times h_{A_n}} & & \downarrow{\scriptstyle h_B} \\ [\![A_1]\!]_{\mathbb{N}} \times \cdots \times [\![A_n]\!]_{\mathbb{N}} & \xrightarrow[[\![f]\!]_{\mathbb{N}}]{} & [\![B]\!]_{\mathbb{N}} \end{array} \tag{MHom-Func}$$

For finite products, given morphisms

$$[\![B]\!]_{\mathbb{M}} \xleftarrow{u} [\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_1]\!]_{\mathbb{M}} \xrightarrow{v} [\![C]\!]_{\mathbb{M}}$$

$$[\![B]\!]_{\mathbb{N}} \xleftarrow{s} [\![A_1]\!]_{\mathbb{N}} \times \cdots \times [\![A_1]\!]_{\mathbb{N}} \xrightarrow{t} [\![C]\!]_{\mathbb{N}}$$

the following diagrams commute

$$\begin{array}{ccc} [\![A]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}} & \xrightarrow{h_{A\times B}} & [\![A]\!]_{\mathbb{N}} \times [\![B]\!]_{\mathbb{N}} \\ \downarrow{\scriptstyle \pi_1} & & \downarrow{\scriptstyle \pi_1} \\ [\![A]\!]_{\mathbb{M}} & \xrightarrow[h_A]{} & [\![A]\!]_{\mathbb{N}} \end{array}$$

$$\begin{array}{ccc} [\![A]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}} & \xrightarrow{h_{A\times B}} & [\![A]\!]_{\mathbb{N}} \times [\![B]\!]_{\mathbb{N}} \\ \downarrow{\scriptstyle \pi_2} & & \downarrow{\scriptstyle \pi_2} \\ [\![B]\!]_{\mathbb{M}} & \xrightarrow[h_B]{} & [\![B]\!]_{\mathbb{N}} \end{array} \tag{MHom-$\times$}$$

$$\begin{array}{ccc} [\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}} & \xrightarrow{h_{A_1\times\cdots\times A_n}} & [\![A_1]\!]_{\mathbb{N}} \times \cdots \times [\![A_n]\!]_{\mathbb{N}} \\ \downarrow{\scriptstyle \langle u,v\rangle} & & \downarrow{\scriptstyle \langle s,t\rangle} \\ [\![B]\!]_{\mathbb{M}} \times [\![C]\!]_{\mathbb{M}} & \xrightarrow[h_{B\times C}]{} & [\![B]\!]_{\mathbb{N}} \times [\![C]\!]_{\mathbb{N}} \end{array}$$

Finally, for function types, given morphisms

$$[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_1]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}} \xrightarrow{f} [\![C]\!]_{\mathbb{M}}$$

$$[\![A_1]\!]_{\mathbb{N}} \times \cdots \times [\![A_1]\!]_{\mathbb{N}} \times [\![B]\!]_{\mathbb{N}} \xrightarrow{g} [\![C]\!]_{\mathbb{N}}$$

the following diagrams commute

$$\begin{array}{ccc}
[\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}} \times [\![A]\!]_{\mathbb{M}} & \xrightarrow{h_{A\Rightarrow B\times A}} & [\![A]\!]_{\mathbb{N}} \Rightarrow [\![B]\!]_{\mathbb{N}} \times [\![A]\!]_{\mathbb{N}} \\
\downarrow{\scriptstyle \mathsf{ev}_{[\![A]\!]_{\mathbb{M}},[\![B]\!]_{\mathbb{M}}}} & & \downarrow{\scriptstyle \mathsf{ev}_{[\![A]\!]_{\mathbb{N}},[\![B]\!]_{\mathbb{N}}}} \\
[\![B]\!]_{\mathbb{M}} & \xrightarrow[h_B]{} & [\![B]\!]_{\mathbb{N}}
\end{array}$$

$$\begin{array}{ccc}
[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}} & \xrightarrow{h_{A_1\times\cdots\times A_n}} & [\![A_1]\!]_{\mathbb{N}} \times \cdots \times [\![A_n]\!]_{\mathbb{N}} \\
\downarrow{\scriptstyle \mathsf{curry}(f)} & & \downarrow{\scriptstyle \mathsf{curry}(g)} \\
[\![B]\!]_{\mathbb{M}} \Rightarrow [\![C]\!]_{\mathbb{M}} & \xrightarrow[h_{B\times C}]{} & [\![B]\!]_{\mathbb{N}} \Rightarrow [\![C]\!]_{\mathbb{N}}
\end{array} \tag{MHom-⇒}$$

Definition 4.11 (Model isomorphism). A *model isomorphism* $h\colon \mathbb{M} \xrightarrow{\sim} \mathbb{N}$ is a model homomorphism where each component h_A is required to be an isomorphism:

$$h_A\colon [\![A]\!]_{\mathbb{M}} \xrightarrow{\sim} [\![A]\!]_{\mathbb{N}}.$$

It is generated by a collection of isomorphisms $h_G\colon [\![G]\!]_{\mathbb{M}} \xrightarrow{\sim} [\![G]\!]_{\mathbb{N}}$ for each ground type $G \in \mathsf{TV}$; h is defined component-wise

$$h_X(X) = \begin{cases} h_G([\![G]\!]_{\mathbb{M}}) & \text{if } X = [\![G]\!]_{\mathbb{M}}, G \in \mathsf{TV}, \\ (h_A \times h_B)([\![A]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}}) & \text{if } X = [\![A \times B]\!]_{\mathbb{M}}, A \times B \in \mathsf{ST}(\mathsf{TV}), \\ (h_A^{-1} \Rightarrow h_B)([\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}}) & \text{if } X = [\![A \to B]\!]_{\mathbb{M}}, A \to B \in \mathsf{ST}(\mathsf{TV}). \end{cases} \tag{MIso-Components}$$

Proposition 4.12. *The definition of model isomorphism suffices to define an isomorphism between two models.*

Proof. See Appendix B, page 46. □

Definition 4.13 (Category of $\lambda^{\to\times}$ models). Given a $\lambda^{\to\times}$-theory $\mathcal{T}$ and a Cartesian closed category $\mathscr{C}$, define

$$\mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{C})$$

to be the category with models of $\mathcal{T}$ in $\mathscr{C}$ as objects, and model isomorphisms as morphisms.

Identities are given by the model isomorphism generated by base components of identity morphisms of $\mathscr{C}$:

$$\mathrm{id}_{\mathbb{M}_G} \coloneqq \mathrm{id}_{[\![G]\!]_{\mathbb{M}}}.$$

Composition between two model isomorphisms is given by the model isomorphism generated by the composites of the base components:

$$(h \circ g)_G \coloneqq h_G \circ g_G.$$

$$\begin{array}{ccc}
B \Rightarrow C & \qquad & B \Rightarrow C \times A \\
\downarrow{\scriptstyle \mathsf{curry}(g\circ \mathsf{ev}_{B,C}\circ \mathrm{id}_{B\Rightarrow C}\times f)} & & \downarrow{\scriptstyle \mathrm{id}_{B\Rightarrow C}\times f} \quad \searrow^{g\circ \mathsf{ev}_{B,C}\circ \mathrm{id}_{B\Rightarrow C}\times f} \\
A \Rightarrow D & & B \Rightarrow C \times B \xrightarrow[\mathsf{ev}_{B,C}]{} C \xrightarrow[g]{} D
\end{array}$$

Definition 4.14 (Model-translating functor). Given a $\lambda^{\to\times}$-theory $\mathcal{T}$, modelled by $\mathbb{M}$ in $\mathscr{C}$, and Cartesian closed functor $F\colon \mathscr{C} \to \mathscr{D}$, we define a functor

$$F_*\colon \mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{C}) \to \mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{D})$$

such that $F_*(\mathbb{M})$ is a model of $\mathcal{T}$ in $\mathscr{D}$.

$F_*(\mathbb{M})$ interprets the ground types $G \in \mathsf{TV}$ by sending the interpretation in $\mathbb{M}$ along F:

$$[\![G]\!]_{F_*(\mathbb{M})} \coloneqq F([\![G]\!]_{\mathbb{M}}); \tag{ModTrans-Type-Base}$$

it also sends the unit type as the terminal object:

$$[\![\mathsf{unit}]\!]_{F_*(\mathbb{M})} \coloneqq 1_{\mathscr{D}}. \tag{ModTrans-Type-unit}$$

By mimicking the categorical semantics for product and function types,

$$[\![A \to B]\!]_{F_*(\mathbb{M})} \coloneqq [\![A]\!]_{F_*(\mathbb{M})} \Rightarrow [\![A]\!]_{F_*(\mathbb{M})}, \tag{ModTrans-Type-$\to$}$$

$$[\![A \times B]\!]_{F_*(\mathbb{M})} \coloneqq [\![A]\!]_{F_*(\mathbb{M})} \times [\![A]\!]_{F_*(\mathbb{M})}, \tag{ModTrans-Type-$\times$}$$

this suffices to give rise to a canonical isomorphism parameterised by a type and a model

$$P_{C\,\mathbb{M}} \colon [\![C]\!]_{F_*(\mathbb{M})} \xrightarrow{\sim} F([\![C]\!]_{\mathbb{M}}), \tag{ModTrans-Type}$$

where $A, B, C \in \mathsf{ST}(\mathsf{TV})$. We omit the model when it is unambiguous from context, writing P_C. At ground types, the canonical choice for this isomorphism is the identity, due to (ModTrans-Type-Base).

Recall that, given a typing context $\Gamma = [x_1{:}A_1, \ldots, x_n{:}A_n]$, a term-in-context $\Gamma \vdash x : B$ is modelled in $\mathbb{M}$ by a morphism: $[\![\Gamma]\!]_{\mathbb{M}} \xrightarrow{[\![\Gamma \vdash x:B]\!]_{\mathbb{M}}} [\![B]\!]_{\mathbb{M}}$; similarly to the case for types, the interpretation of such a term in $F_*(\mathbb{M})$ is given by

$$\begin{array}{rcl} & & [\![\Gamma]\!]_{F_*(\mathbb{M})} \qquad\qquad [\![B]\!]_{F_*(\mathbb{M})} \\ & & \quad {\scriptstyle P_{A_1}\times\cdots\times P_{A_n}}\downarrow{\scriptstyle \wr} \qquad\qquad \uparrow{\scriptstyle \wr}\, {\scriptstyle P_B^{-1}} \\ [\![\Gamma \vdash x : B]\!]_{F_*(\mathbb{M})} & \coloneqq & F([\![A_1]\!]_{\mathbb{M}}) \times \cdots \times F([\![A_n]\!]_{\mathbb{M}}) \\ & & \quad {\scriptstyle \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}}}\downarrow{\scriptstyle \wr} \\ & & F([\![A_1 \times \cdots \times A_n]\!]_{\mathbb{M}}) \xrightarrow{F([\![\Gamma \vdash x:B]\!]_{\mathbb{M}})} F([\![B]\!]_{\mathbb{M}}) \end{array} \tag{ModTrans-Term}$$

Functions symbols are interpreted similarly: given $f \colon A_1 \times \cdots \times A_n \to B$ interpreted in $\mathbb{M}$ by $[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}} \xrightarrow{[\![f]\!]_{\mathbb{M}}} [\![B]\!]_{\mathbb{M}}$, we then have

$$\begin{array}{rcl} & & [\![A_1]\!]_{F_*(\mathbb{M})} \times \cdots \times [\![A_n]\!]_{F_*(\mathbb{M})} \qquad [\![B]\!]_{F_*(\mathbb{M})} \\ & & \quad {\scriptstyle P_{A_1}\times\cdots\times P_{A_n}}\downarrow{\scriptstyle \wr} \qquad\qquad \uparrow{\scriptstyle \wr}\, {\scriptstyle P_B^{-1}} \\ [\![f]\!]_{F_*(\mathbb{M})} & \coloneqq & F([\![A_1]\!]_{\mathbb{M}}) \times \cdots \times F([\![A_n]\!]_{\mathbb{M}}) \\ & & \quad {\scriptstyle \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}}}\downarrow{\scriptstyle \wr} \\ & & F([\![A_1 \times \cdots \times A_n]\!]_{\mathbb{M}}) \xrightarrow{F([\![f]\!]_{\mathbb{M}})} F([\![B]\!]_{\mathbb{M}}) \end{array} \tag{ModTrans-FSym}$$

as

$$\begin{array}{ll} & [\![A_1]\!]_{F_*(\mathbb{M})} \times \cdots \times [\![A_n]\!]_{F_*(\mathbb{M})} \\ \cong & \quad \{\ \text{(ModTrans-Type)}\ \} \\ & F([\![A_1]\!]_{\mathbb{M}}) \times \cdots \times F([\![A_n]\!]_{\mathbb{M}}) \\ \cong & \quad \{\ F \text{ preserves finite products along } \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}}\ \} \end{array}$$

$$
\begin{aligned}
& F(\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}) \\
= \quad & \{ \text{ (CatSem-}\times\text{) } \} \\
& F(\llbracket A_1 \times \cdots \times A_n \rrbracket_{\mathbb{M}}) \\
\cong \quad & \{ \text{ (ModTrans-Type) } \} \\
& \llbracket A_1 \times \cdots \times A_n \rrbracket_{F_*(\mathbb{M})}.
\end{aligned}
$$

A model $\mathbb{M}$ of a $\lambda^{\to\times}$-theory in $\mathscr{C}$ is a collection of objects and morphisms of $\mathscr{C}$, so the object-mapping part of the functor does indeed map a collection of objects and morphisms of $\mathscr{C}$ to a collection of objects and morphisms in $\mathscr{D}$.

To complete the definition of F_*, we must define its action on morphisms of $\mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{C})$: model isomorphisms. The model isomorphism $h\colon \mathbb{M} \xrightarrow{\sim} \mathbb{N}$ is given by its components which map interpretations of ground types, so we define $F_*(h)$ to be the model isomorphism with base components

$$(F_*(h))_G \coloneqq F(h_G)\colon \llbracket G \rrbracket_{F_*(\mathbb{M})} \to \llbracket G \rrbracket_{F_*(\mathbb{N})} \qquad \text{(ModTrans-MIso-Base)}$$

for each ground type $G \in \mathsf{TV}$.

F_* can be thought of as a free functor generated by a $\lambda^{\to\times}$-theory model $\mathbb{M}$ and a Cartesian closed functor F.

Proposition 4.15. *The interpretation of a type in $F_*(\mathbb{M})$ is isomorphic to the image of its interpretation in $\mathbb{M}$ in F* (ModTrans-Type).

Proof. See Appendix B, page 47. □

Proposition 4.16. *Given a model $\mathbb{M}$ of $\mathcal{T} = (\sigma, \mathcal{A})$ in $\mathscr{C}$, $F_*(\mathbb{M})$ is a well-defined model of $\mathcal{T}$ in $\mathscr{D}$.*

Proof. See Appendix B, page 47. □

Now we prove a few lemmas regarding the interaction between the canonical isomorphism P and product and exponential preservation isomorphisms Φ and Ψ.

Lemma 4.17. *For all $A_i \in \mathsf{ST}(\mathsf{TV})$ such that $i > 0$,*

1. $P_{A_1 \times \cdots \times A_n} = \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$;
2. $P^{-1}_{A_1 \times \cdots \times A_n} = P^{-1}_{A_1} \times \cdots \times P^{-1}_{A_n} \circ \Phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}}$.

Proof. By induction over n.

$n = 1$ **case** Trivial (recall that $\Phi_X = \mathrm{id}_X$ if X is not a product). ◁

$n = k + 1$ **case** By diagram chase:

$$
\begin{array}{ccc}
\llbracket A_1 \times \cdots \times A_k \times A_n \rrbracket_{F_*(\mathbb{M})} & \xrightarrow[P_{A_1 \times \cdots \times A_k \times A_n}]{\sim} & F(\llbracket A_1 \times \cdots \times A_k \times A_n \rrbracket_{\mathbb{M}}) \\
\Big\| \text{ Proposition 4.16} & & \Big\| \\
\Big\| & & \Big\| \\
\llbracket A_1 \times \cdots \times A_k \rrbracket_{F_*(\mathbb{M})} \times \llbracket A_n \rrbracket_{F_*(\mathbb{M})} & & \mathbb{M} \text{ models } \mathcal{T} \\
\Big\downarrow \scriptstyle P_{A_1 \times \cdots \times A_k} \times P_{A_n} & & \Big\| \\
F(\llbracket A_1 \times \cdots \times A_k \rrbracket_{\mathbb{M}}) \times F(\llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow[\Phi^{-1}_{\llbracket A_1 \times \cdots \times A_k \rrbracket_{\mathbb{M}} \times \llbracket A_n \rrbracket_{\mathbb{M}}}]{\sim} & F(\llbracket A_1 \times \cdots \times A_k \rrbracket_{\mathbb{M}} \times \llbracket A_n \rrbracket_{\mathbb{M}})
\end{array}
$$

2 is proven similarly by inverting all of the morphisms. ◁

□

Lemma 4.18. *For all $A, B \in \mathsf{ST}(\mathsf{TV})$,*

1. $P_{A\to B} = \Psi^{-1}_{[\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}}} \circ P_A^{-1} \Rightarrow P_B$;
2. $P^{-1}_{A\to B} = P_A \Rightarrow P_B^{-1} \circ \Psi_{[\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}}}$.

Proof. By diagram chase:

$$\begin{array}{ccc}
[\![A \to B]\!]_{F_*(\mathbb{M})} & \xrightarrow[P_{A\to B}]{\sim} & F([\![A \to B]\!]_{\mathbb{M}}) \\
\| & & \| \\
\text{Proposition 4.16} & & \\
\| & & \| \\
[\![A]\!]_{F_*(\mathbb{M})} \Rightarrow [\![B]\!]_{F_*(\mathbb{M})} & & \mathbb{M} \text{ models } \mathcal{T} \\
\downarrow \sim & & \| \\
P_A^{-1} \Rightarrow P_B & & \| \\
\downarrow & & \| \\
F([\![A]\!]_{\mathbb{M}}) \Rightarrow F([\![B]\!]_{\mathbb{M}}) & \xrightarrow[\Phi^{-1}_{[\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}}}]{\sim} & F([\![A]\!]_{\mathbb{M}} \Rightarrow [\![B]\!]_{\mathbb{M}})
\end{array}$$

Observe that $(P_A^{-1} \Rightarrow P_B)^{-1} = P_A \Rightarrow P_B^{-1}$; then 2 can be proven by inverting all of the morphisms.

□

Next, we consider model isomorphisms.

Proposition 4.19. *Given a model isomorphism $h\colon \mathbb{M} \xrightarrow{\sim} \mathbb{N}$, for all $A \in \mathsf{ST}(\mathsf{TV})$,*

$$F_*(h)_A\colon [\![A]\!]_{F_*(\mathbb{M})} \xrightarrow{\sim} [\![A]\!]_{F_*(\mathbb{N})} = P^{-1}_{A\,\mathbb{N}} \circ F(h_A) \circ P_{A\,\mathbb{M}}. \tag{ModTrans-MIso}$$

Proof. See Appendix B, page 52. □

Proposition 4.20. *F_* is a functor.*

Proof. We must show that F_* preserves

1. domains and codomains: $F_*(h\colon \mathbb{M} \to \mathbb{N}) = F_*(h)\colon F_*(\mathbb{M}) \to F_*(\mathbb{N})$;
2. identities: $F_*(\mathrm{id}_{\mathbb{M}}) = \mathrm{id}_{F_*(\mathbb{M})}$;
3. composition: $F_*(g \circ h) = F_*(g) \circ F_*(h)$.

The first property is follows from (ModTrans-MIso-Base); each base component of $F_*(h)$, $F_*(h)_G$, is equal to $F(h_G)\colon [\![G]\!]_{F_*(\mathbb{M})} \to [\![G]\!]_{F_*(\mathbb{N})}$, so by (MIso-Components) $F_*(h)$ is generated as some mapping from model $F_*(\mathbb{M})$ to $F_*(\mathbb{N})$. For the second property, observe that $F_*(\mathrm{id}_{\mathbb{M}})$ has base components $F(\mathrm{id}_{[\![G]\!]_{\mathbb{M}}})$ for each ground type G, which is equal to $\mathrm{id}_{[\![G]\!]_{F_*(\mathbb{M})}}$ by functoriality of F, and so this does define an identity morphism of $F_*(\mathbb{M})$. Finally, for the third property, we get that $F_*(g \circ h)$ is a model isomorphism given by base components $F((g \circ h)_G) = F(g_G \circ h_G) = F(g_G) \circ F(h_G)$ by functoriality, and so is the composition of model isomorphisms $F_*(g) \circ F_*(h)$ as required.

□

This shows that F_* type checks, but it remains to verify that $F_*(h)$ is a valid model isomorphism.

Proposition 4.21. *Given a model isomorphism $h\colon \mathbb{M} \to \mathbb{N}$, $F_*(h)\colon F_*(\mathbb{M}) \to F_*(\mathbb{N})$ is a model isomorphism.*

Proof. See Appendix B, page 53. □

Definition 4.22 (Model isomorphism freely generated by natural isomorphism). If $F\colon \mathscr{C} \to \mathscr{D}$ and $E\colon \mathscr{C} \to \mathscr{D}$ are Cartesian closed functors, with a natural isomorphism $\phi\colon F \tilde{\Rightarrow} E$, then we define the model isomorphism

$$\phi_*\mathbb{M}\colon F_*(\mathbb{M}) \xrightarrow{\sim} E_*(\mathbb{M})$$

generated by the collection of isomorphisms

$$(\phi_*\mathbb{M})_G \coloneqq \phi_{[\![G]\!]_{\mathbb{M}}}\colon F([\![G]\!]_{\mathbb{M}}) = [\![G]\!]_{F_*(\mathbb{M})} \xrightarrow{\sim} [\![G]\!]_{E_*(\mathbb{M})} = E([\![G]\!]_{\mathbb{M}})$$

for each ground type $G \in \mathsf{TV}$ of a $\lambda^{\to\times}$-theory.

Proposition 4.23. *$\phi_*\mathbb{M}$ is a model isomorphism.*

Proof. See Appendix B, page 54. □

Now we combine the machinery together.

4.2.3 Categorical equivalence.

Definition 4.24 (Modelling functor). If $\mathscr{C}$ and $\mathscr{D}$ are Cartesian closed categories, $\mathcal{T}$ is a $\lambda^{\to\times}$-theory, and $\mathbb{M}$ a model of $\mathcal{T}$ in $\mathscr{C}$, then we define the *family of modelling functors*

$$\mathbf{Ap}_{\mathbb{M}}\colon \mathbf{CCCat}_{\simeq}(\mathscr{C}, \mathscr{D}) \to \mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{D})$$

which maps Cartesian closed functors to models:

$$F \mapsto F_*(\mathbb{M}),$$

and natural isomorphisms to model isomorphisms:

$$\phi \mapsto \phi_*\mathbb{M}.$$

Definition 4.25 (Generic model). A *generic* model of a $\lambda^{\to\times}$-theory $\mathcal{T}$ is a model $\mathbb{G}$ such that the modelling functor

$$\mathbf{Ap}_{\mathbb{G}}\colon \mathbf{CCCat}_{\simeq}\left(\mathbf{Syn}(\mathcal{T}), \mathscr{D}\right) \to \mathbf{Mod}_{\simeq}(\mathcal{T}, \mathscr{D})$$

is fully faithful and essentially surjective (i.e. an equivalence).

The canonical choice for $\mathbb{G}$ in $\mathbf{Syn}(\mathcal{T})$ is given by associating each ground type $G \in \mathsf{TV}$ to its syntactic object:

$$[\![G]\!]_{\mathbb{G}} \coloneqq G,$$

and each constant $c{:}A$ and n-ary function symbol $f{:}A_1 \times \cdots \times A_n \to B$ to its respective morphism:

$$[\![c]\!]_{\mathbb{G}} \coloneqq [\![\ulcorner c \urcorner^A]\!]_{\mathcal{T}},$$
$$[\![f]\!]_{\mathbb{G}} \coloneqq [\![\ulcorner f \urcorner^{A_1\times\cdots\times A_n \to B}]\!]_{\mathcal{T}}.$$

The definition of $\mathbf{Syn}(\mathcal{T})$ then ensures that $[\![A]\!]_{\mathbb{G}} = A$ for all $A \in \mathsf{ST}(\mathsf{TV})$, each term-in-context is associated to it its $\mathcal{T}$-equivalence class, and typing contexts are also modelled in the ordinary way (CatSem-Γ).

In fulfilling Definition 3.1, we have shown that $\mathbb{G}$ is a structure of σ, and hence by the Soundness Theorem it is a model.

$\mathbf{Ap}_{\mathbb{G}}$ sends the identity functor $\mathrm{id}_{\mathbf{Syn}(\mathcal{T})}$ to $\mathbb{G}$.

Such a model is generic, in the sense that we make no arbitrary choices: merely construct the syntactic category and assign interpretations to their syntactic duals. Generic models are important because they capture in the most precise fashion the notion of 'minimal' with respect to models.

Definition 4.26 (Classifying category). **Syn** $(\mathcal{T})$ is described as *classifying* if its generic model exists, along with an equivalence

$$\mathbf{CCCat}_{\simeq}\left(\mathbf{Syn}\left(\mathcal{T}\right), \mathscr{D}\right) \simeq \mathbf{Mod}_{\simeq}\left(\mathcal{T}, \mathscr{D}\right)$$

for all Cartesian closed categories $\mathscr{D}$.

To demonstrate such an equivalence, we can give witness to functors in either direction such that the compositions are naturally isomorphic to the relevant identity functor.

Definition 4.27 (Generic modelling functor). The *generic modelling functor* is the modelling functor parameterised by the generic model $\mathbb{G}$,

$$\mathbf{Ap}_{\mathbb{G}} \colon \mathbf{CCCat}_{\simeq}\left(\mathbf{Syn}\left(\mathcal{T}\right), \mathscr{D}\right) \to \mathbf{Mod}_{\simeq}\left(\mathcal{T}, \mathscr{D}\right).$$

This functor maps each Cartesian closed functor $F \colon \mathbf{Syn}\left(\mathcal{T}\right) \to \mathscr{D}$ to a generic model $F_*(\mathbb{G})$.
Now we define its 'inverse'.

Definition 4.28 (Inverse generic modelling functor). The *inverse generic modelling functor* is given by

$$\mathbf{Ap}_{\mathbb{G}}^{-1} \colon \mathbf{Mod}_{\simeq}\left(\mathcal{T}, \mathscr{D}\right) \to \mathbf{CCCat}_{\simeq}\left(\mathbf{Syn}\left(\mathcal{T}\right), \mathscr{D}\right)$$

which acts on models $\mathbb{M}$ of $\mathcal{T}$ in $\mathscr{D}$, yielding a Cartesian closed functor $F \colon \mathbf{Syn}\left(\mathcal{T}\right) \to \mathscr{D}$, which maps syntactic objects to the interpretation of their corresponding type in $\mathbb{M}$:

$$A \mapsto \left[\!\left[A\right]\!\right]_{\mathbb{M}},$$

and $\mathcal{T}$-equivalence classes of terms to their interpretation in $\mathbb{M}$:

$$\left[\ulcorner t \urcorner^{A}\right]_{\mathcal{T}} \mapsto \left[\!\left[\vdash t : A\right]\!\right]_{\mathbb{M}}.$$

Well-definedness arises from the Soundness Theorem, and it is clear that this is a (strict) Cartesian closed functor.

The action of $\mathbf{Ap}_{\mathbb{G}}^{-1}$ on a model isomorphism $h \colon \mathbb{M} \to \mathbb{N}$ is a natural isomorphism between functors $\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})$ and $\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{N})$, with components

$$\left(\mathbf{Ap}_{\mathbb{G}}^{-1}(h)\right)_{A} : \mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})(A) = \left[\!\left[A\right]\!\right]_{\mathbb{M}} \xrightarrow{\sim} \left[\!\left[A\right]\!\right]_{\mathbb{N}} = \mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{N})(A) \coloneqq h_A.$$

We prove that this really defines a natural isomorphism in the following proposition.

Proposition 4.29. *Given a model isomorphism $h \colon \mathbb{M} \to \mathbb{N}$, $\mathbf{Ap}_{\mathbb{G}}^{-1}(h)$ is a natural isomorphism.*

Proof. See Appendix B, page 55. □

Theorem 4.30 (**Syn** $(\mathcal{T})$ is classifying). *The syntactic category* **Syn** $(\mathcal{T})$ *of a $\lambda^{\to\times}$-theory is classifying.*

Proof. **Syn** $(\mathcal{T})$ contains a generic model $\mathbb{G}$ of $\mathcal{T}$, and we give witness to the categorical equivalence equivalence with natural isomorphisms

$$\epsilon \colon \mathbf{Ap}_{\mathbb{G}} \circ \mathbf{Ap}_{\mathbb{G}}^{-1} \xRightarrow{\sim} \mathrm{id}_{\mathbf{Mod}_{\simeq}(\mathcal{T},\mathscr{D})} \quad \text{and} \quad \eta \colon \mathbf{Ap}_{\mathbb{G}}^{-1} \circ \mathbf{Ap}_{\mathbb{G}} \xRightarrow{\sim} \mathrm{id}_{\mathbf{CCCat}_{\simeq}\left(\mathbf{Syn}(\mathcal{T}),\mathscr{D}\right)}.$$

We define ϵ component-wise on models $\mathbb{M}$ of $\mathcal{T}$ in a Cartesian closed category $\mathscr{D}$:

$$\epsilon_{\mathbb{M}} \colon \mathbf{Ap}_{\mathbb{G}}(\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})) \to \mathbb{M}$$

is required to be a model isomorphism, and observe that $\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})$ yields a Cartesian closed functor $F\colon \mathbf{Syn}(\mathcal{T}) \to \mathscr{D}$ which maps syntax to its $\mathbb{M}$-interpretation. Now, $\mathbf{Ap}_{\mathbb{G}}(F)$ is the generic model $F_*(\mathbb{G})$. For ground types $G \in \mathsf{TV}$, recall from (ModTrans-Type-Base) that $F_*(\mathbb{G})$ interprets G in $\mathbb{G}$ through F:

$$[\![G]\!]_{F_*(\mathbb{G})} = F([\![G]\!]_{\mathbb{G}}) = F(G) = [\![G]\!]_{\mathbb{M}}.$$

It suffices to define the model isomorphism by giving its base isomorphisms at ground types; let these be identity isomorphisms

$$(\epsilon_{\mathbb{M}})_G \coloneqq \mathrm{id}_{[\![G]\!]_{\mathbb{M}}}\colon [\![G]\!]_{\mathbf{Ap}_{\mathbb{G}}(\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M}))} = [\![G]\!]_{\mathbb{M}} \to [\![G]\!]_{\mathbb{M}}.$$

Now for the reverse direction, given a Cartesian closed functor $F\colon \mathbf{Syn}(\mathcal{T}) \to \mathscr{D}$, $\mathbf{Ap}_{\mathbb{G}}(F)$ is the generic model $F_*(\mathbb{G})$, so $\mathbf{Ap}_{\mathbb{G}}^{-1}(F_*(\mathbb{G}))$ gives a Cartesian closed functor $\mathbf{Syn}(\mathcal{T}) \to \mathscr{D}$ mapping

$$A \mapsto [\![A]\!]_{F_*(\mathbb{G})} = F([\![A]\!]_{\mathbb{G}}) = F(A),$$

so we define η_F to be the identity natural isomorphism, with components

$$(\eta_F)_A \coloneqq \mathrm{id}_{F(A)}\colon F(A) \to F(A).$$

□

This result is the crux of the categorical type theory correspondence, effectively stating that $\lambda^{\to\times}$-theories and Cartesian closed categories are notionally the same data.

Due to the strictness of $\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})$, we can prove an even stronger property.

Theorem 4.31 (Universal property of generic model). *For all models $\mathbb{M}$ of a $\lambda^{\to\times}$-theory $\mathcal{T}$ in a Cartesian closed category $\mathscr{C}$, there exists a Cartesian closed functor $F\colon \mathbf{Syn}(\mathcal{T}) \to \mathscr{C}$ such that $F_*(\mathbb{G}) = \mathbb{M}$. Such an F is unique up to (canonical) isomorphism.*

In other words, a $\lambda^{\to\times}$-model of $\mathcal{T}$ in a Cartesian closed category $\mathscr{C}$ is precisely characterised by a unique functor from $\mathbf{Syn}(\mathcal{T})$ to $\mathscr{C}$.

Proof. Take F to be $\mathbf{Ap}_{\mathbb{G}}^{-1}(\mathbb{M})$. For all simple types A of $\mathcal{T}$,

$$[\![A]\!]_{F_*(\mathbb{G})} = F([\![A]\!]_{\mathbb{G}}) = F(A) = [\![A]\!]_{\mathbb{M}}.$$

Similarly, for all closed terms $\vdash t : A$,

$$[\![\vdash t : A]\!]_{F_*(\mathbb{G})} = F([\![\vdash t : A]\!]_{\mathbb{G}}) = F([\![\ulcorner t \urcorner^A]\!]_{\mathcal{T}}) = [\![\vdash t : A]\!]_{\mathbb{M}}.$$

The first equalities arise from Proposition 4.15, but strictness of F ensures that the isomorphism is an equality.

Suppose $G\colon \mathbf{Syn}(\mathcal{T}) \to \mathscr{C}$ is a Cartesian closed functor such that $G_*(\mathbb{G}) = \mathbb{M}$ — we will show that $F(X) \cong G(X)$ necessarily: let A be an object of $\mathbf{Syn}(\mathcal{T})$; then

$$F(A) = [\![A]\!]_{\mathbb{M}} = [\![A]\!]_{G_*(\mathbb{G})} \cong G([\![A]\!]_{\mathbb{G}}) = G(A).$$

Moreover, Proposition 4.15 constructs the isomorphism, so this induces a natural isomorphism and hence $F \cong G$.

□

Although we only defined $\mathbf{Ap}_{\mathbb{G}}^{-1}$, it can be generalised to other models $\mathbb{M}$; however, note that $\mathbf{Ap}_{\mathbb{M}}(\mathbf{Ap}_{\mathbb{M}}^{-1}(\mathbb{N}))$ is not necessarily equal to $\mathbb{N}$: the corresponding objects of the structures are merely required to be isomorphic to one another, except at the interpretations of ground types where strict equality is required by (ModTrans-Type-Base). This occurs when the image of $\mathbf{Ap}_{\mathbb{M}}^{-1}$ is a non-strict Cartesian closed functor.

Proposition 4.32. *Every Cartesian closed category is equivalent to the syntactic category of its internal language:*

$$\mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right) \cong \mathscr{C}.$$

Proof. See Appendix B, page 60. □

At last we can prove the main result of this section.

Theorem 4.33 (Completeness of categorical semantics). *Given a $\lambda^{\to\times}$-theory $\mathcal{T}$, there exists a unique (up to equivalence) category containing a model which minimally models $\mathcal{T}$; that is, for terms-in-context $\Gamma \vdash u : A$ and $\Gamma \vdash v : A$,*

$$\Gamma \vdash u = v : A \iff [\![\Gamma \vdash u : A]\!] = [\![\Gamma \vdash v : A]\!].$$

Proof. For existence, consider the classifying category $\mathbf{Syn}(\mathcal{T})$, and the model it contains is the generic model $\mathbb{G}$. To see that it satisfies the required property, first we must model $\Gamma \vdash u : A$ in $\mathbf{Syn}(\mathcal{T})$, associating to $[\![\Gamma \vdash u : A]\!]$ some morphism $[\![\Gamma]\!] = B_1 \times \cdots \times B_n \to A = [\![A]\!]$ where $\Gamma = [x_1{:}B_1, \ldots, x_n{:}B_n]$. From $\Gamma \vdash u : A$, repeatedly use the abs rule to give a closed term which can be associated to the morphism $[\![u]\!] = [\ulcorner \lambda x_1{:}B_1. \ldots \lambda x_n{:}B_n.u^{\ulcorner B_1 \to \cdots \to B_n \to A}]_{\mathcal{T}} : 1 = \mathsf{unit} \to B_1 \Rightarrow \cdots \Rightarrow B_n \Rightarrow A$. Now, uncurry $[\![u]\!]$ until the it has a codomain A, yielding $[\![u]\!]' = \mathsf{uncurry}\left(\ldots \mathsf{uncurry}\left([\![u]\!]\right)\right) : \mathsf{unit} \times B_1 \times \cdots \times B_n \to A$, which can be composed with the canonical isomorphisms induced by $1 \times X = \mathsf{unit} \times X \xrightarrow{\sim} X$ and $(X \times Y) \times Z \xrightarrow{\sim} X \times (Y \times Z)$ for all objects X, Y, and Z giving

$$[\![\Gamma \vdash u : A]\!] = [\![\Gamma]\!] = B_1 \times \cdots \times B_n \xrightarrow{\sim} \mathsf{unit} \times (B_1 \times \cdots \times B_n) \xrightarrow{\sim} \mathsf{unit} \times B_1 \times \cdots \times B_n \xrightarrow{[\![u]\!]'} A.$$

$\Gamma \vdash v : A$ is modelled similarly. By assumption, $[\![\Gamma \vdash u : A]\!] = [\![\Gamma \vdash v : A]\!]$, and so $[\![u]\!]' = [\![v]\!]'$. Hence $[\![u]\!] = [\![v]\!]$, from which we can infer $\vdash \lambda x_1{:}B_1. \ldots \lambda x_n{:}B_n.u = \lambda x_1{:}B_1. \ldots \lambda x_n{:}B_n.v : B_1 \to \cdots \to B_n \to A$. But from the structure, such a derivation must have arisen via repeated application of the abs rule from a derivation of $[x_1{:}B_1, \ldots, x_n{:}B_n] = \Gamma \vdash u = v : A$, which is exactly what was required.

To show uniqueness (up to equivalence), let $\mathscr{C}$ be another category which satisfies the property; we will show that it is equivalent to $\mathbf{Syn}(\mathcal{T})$. $\mathscr{C}$ minimally models $\mathcal{T}$, so its internal language is $\mathcal{T}$:

$$\mathcal{T} = \mathbf{Lan}\left(\mathscr{C}\right).$$

By Proposition 4.32, we deduce that $\mathscr{C} \cong \mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right) = \mathbf{Syn}\left(\mathcal{T}\right)$ as required.

□

This concludes our development of the completeness of categorical semantics for $\lambda^{\to\times}$-theories. For a similar but far more terse approach, whereby $\lambda^{\to\times}$-theories themselves form categories under a suitable notion of translation, consult Awodey & Bauer (2017 sec. 2.4).

5 CONCLUSION

We have shown the precise way in which simply-typed λ-calculi and Cartesian closed categories are 'the same', namely that any $\lambda^{\to\times}$-theory can be modelled by any Cartesian closed category, and all models uniquely factor through the generic model arising from the syntactic category, which is classifying. Such a model is unique up to categorical equivalence, which is the appropriate notion of equality in the context of categories. The constructions we have presented are entirely constructive, and yield a canonical way to transform between $\lambda^{\to\times}$-theories and Cartesian closed categories: we have shown how to freely construct the syntactic category from a $\lambda^{\to\times}$-theory, and how to generate the internal language of a Cartesian closed category, which is guaranteed to be a $\lambda^{\to\times}$-theory.

5.1 Further work

Using this work as a base template for categorical semantics semantics of λ-calculi, we can pursue extensions in many directions. For richer type theories, like the polymorphic λ-calculus and dependently-typed λ-calculi, we can explore their categorical setting; unlike the case of simple-types, there are many distinct approaches to this. Polymorphic λ-calculus is treated by Crole (1993 Chapters 5–6) in a similar spirit to this dissertation, and Jacobs (1999 Chapter 8) provides an account using fibred category theory. For dependent type theories alone, Pitts (2001 p. 62) lists 9 different accounts, and the dust has still not settled on which approach is definitive. Another interesting direction is the restriction of the simply-typed λ-calculus to type theories corresponding to substructural fragments of propositional logic: for instance, the linear λ-calculus, which can elegantly express various safety properties that programmers will find useful, relates to symmetric monoidal categories in the same way (Abramsky & Tzevelekos 2010 p. 77).

This is still confining our attention on one edge of the Curry-Howard-Lambek correspondence, and the other edges capture a wealth of information too much to adequately summarise.

> *In my view, what makes something a scientific discovery — in this realm — is you know that it makes sense from all three points of view, and if you have a concept that makes sense from all three points of view, it's a permanent advance of the human intellect.*
>
> Robert Harper (2013) on computational trinitarianism

A CARTESIAN CLOSED CATEGORIES EQUATIONALLY

In this section we make explicit the formulation of 'Cartesian closed category', adapted from Awodey (2010 pp. 134–135).

Definition A.1 (Equational definition of a Cartesian closed category). A category $\mathscr{C}$ is a Cartesian closed category if and only if it has the following structure:

- A distinct object object 1, and for each object X there is a given morphism
$$!_X \colon X \to 1$$
such that for each morphism $f \colon X \to 1$
$$f = !_X.$$
- For each pair of objects X, Y, there is a given object $X \times Y$ and morphisms
$$\pi_1 \colon X \times Y \to X \quad \text{and} \quad \pi_2 \colon X \times Y \to Y$$
and for each pair of morphisms $f \colon Z \to X$ and $g \colon Z \to Y$, there is a given morphism,
$$\langle f, g \rangle \colon Z \to X \times Y$$
such that
$$\begin{aligned} \pi_1 \circ \langle f, g \rangle &= f \\ \pi_2 \circ \langle f, g \rangle &= g \\ \langle \pi_2 \circ h, \pi_2 \circ h \rangle &= h \qquad \text{for all } h \colon Z \to X \times Y. \end{aligned}$$
- For each pair of objects X, Y, there is a given object $X \Rightarrow Y$ and a morphism
$$\mathsf{ev}_{X,Y} \colon X \Rightarrow Y \times X \to Y$$
and for each morphism $f \colon Z \times X \to Y$, there is a given morphism
$$\mathsf{curry}\left(f\right) : Z \to X \Rightarrow Y$$
such that
$$\mathsf{ev}_{X,Y} \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X = f \qquad \text{(curry-}\exists\text{)}$$
and
$$\mathsf{curry}\left(\mathsf{ev}_{X,Y} \circ g \times \mathrm{id}_X\right) = g \qquad \text{(curry-!)}$$
for all $g \colon Z \to X \Rightarrow Y$. We further define
$$\mathsf{uncurry}\left(g\right) \coloneqq \mathsf{ev}_{X,Y} \circ g \times \mathrm{id}_X.$$
- For any morphisms $f \colon X \to U$ and $g \colon Y \to V$, we write[13]
$$\begin{aligned} f \times g &\coloneqq \langle f \circ \pi_1, g \circ \pi_2 \rangle \colon X \times Y \to U \times V, \\ f \Rightarrow g &\coloneqq \mathsf{curry}\left(g \circ \mathsf{ev}_{U,Y} \circ \mathrm{id}_{U \Rightarrow Y} \times f\right) : U \Rightarrow Y \to X \Rightarrow V. \end{aligned}$$

Examining this, it is clear the first two points respectively assert the existence of a terminal object and binary products; equivalently, $\mathscr{C}$ has finite products.

(curry-$\exists$) stipulates the existence of the 'currying' of every appropriately typed morphism. Using our definition for uncurry in (curry-$\exists$) yields
$$f = \mathsf{uncurry}\left(\mathsf{curry}\left(f\right)\right),$$

[13] Formally, this is the action of the bifunctors $- \times -$ and $- \Rightarrow -$ on morphisms.

and in (curry-!) yields

$$g = \mathsf{curry}\left(\mathsf{uncurry}\left(g\right)\right),$$

so (curry-!) specifies a uniqueness condition. Together, these equations are precisely the universal property of exponential.

In the literature, Cartesian closed categories are often succinctly presented differently by means of adjunctions, which is a very popular *style* of Category Theory. The following definition is from Leinster (2016 p. 165).

Definition A.2 (Adjunctional definition of a Cartesian closed category). A category $\mathscr{C}$ is Cartesian closed if it has finite products, and for each object Z in $\mathscr{C}$, the functor $- \times Z \colon \mathscr{C} \to \mathscr{C}$ has a right adjoint.

Such an adjoint is the exponential functor $Z \Rightarrow - \colon \mathscr{C} \to \mathscr{C}$, and the consequence of the adjunction is that for all objects X, Y, and Z of $\mathscr{C}$,

$$\mathscr{C}(X \times Z, Y) \cong \mathscr{C}(X, Z \Rightarrow Y)$$

naturally in X and Y.

We will make use of the following lemma in proving that the two definitions are equivalent.

Lemma A.3. *In an equational Cartesian closed category, for all morphisms $f \colon X \times Z \to Y$,*

$$\mathsf{curry}\left(f\right) = \mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X \times Z}\right).$$

Proof. Expand

$$\begin{aligned}
& \mathrm{id}_Z \Rightarrow f \\
& \mathsf{curry}\left(f \circ \mathsf{ev}_{Z, X \times Z} \circ \mathrm{id}_{Z \Rightarrow X \times Z} \times \mathrm{id}_Z\right) \\
= \quad & \{ \text{ bifunctoriality of } - \times - \ \} \\
& \mathsf{curry}\left(f \circ \mathsf{ev}_{Z, X \times Z} \circ \mathrm{id}_{Z \Rightarrow X \times Z \times Z}\right) \\
= \quad & \{ \text{ identity cancels through composition } \} \\
& \mathsf{curry}\left(f \circ \mathsf{ev}_{Z, X \times Z}\right),
\end{aligned}$$

and observe that

$$\begin{aligned}
& \mathsf{uncurry}\left(\mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X \times Z}\right)\right) \\
= \quad & \{ \ \mathsf{uncurry} \ \} \\
& \mathsf{ev}_{Z,Y} \circ (\mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X \times Z}\right)) \times \mathrm{id}_Z \\
= \quad & \{ \ - \times Z \text{ functor } \} \\
& \mathsf{ev}_{Z,Y} \circ \mathrm{id}_Z \Rightarrow f \times \mathrm{id}_Z \circ \mathsf{curry}\left(\mathrm{id}_{X \times Z}\right) \times \mathrm{id}_Z \\
= \quad & \{ \ \text{above} \ \} \\
& \mathsf{ev}_{Z,Y} \circ \mathsf{curry}\left(f \circ \mathsf{ev}_{Z, X \times Z}\right) \times \mathrm{id}_Z \circ \mathsf{curry}\left(\mathrm{id}_{X \times Z}\right) \times \mathrm{id}_Z \\
= \quad & \{ \ (\text{curry-}\exists) \ \}
\end{aligned}$$

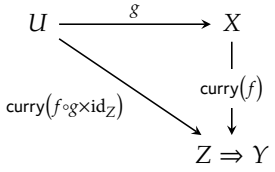
$$
\begin{aligned}
& f \circ \mathsf{ev}_{Z,X\times Z} \circ \mathsf{curry}\left(\mathrm{id}_{X\times Z}\right) \times \mathrm{id}_Z \\
= \quad & \{\ (\mathsf{curry}\text{-}\exists)\ \} \\
& f \circ idX \times Z \\
= \quad & \{\ \text{identity cancels through composition}\ \} \\
& f \\
\Longrightarrow \quad & \{\ \text{apply curry to both sides}\ \} \\
= \quad & \mathsf{curry}\left(f\right) \\
& \mathsf{curry}\left(\mathsf{uncurry}\left(\mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X\times Z}\right)\right)\right) \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& \mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X\times Z}\right).
\end{aligned}
$$

□

It is not immediately obvious how the adjunction relates to the equational definition, but they are indeed equivalent; explicitly, the adjunction stipulates an isomorphism between hom-sets

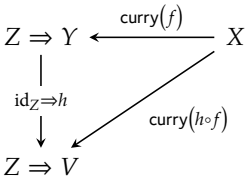
$$\mathsf{uncurry}\left(g\right) = f\colon X \times Z \to Y \mathrel{\tilde{\mapsto}} \mathsf{curry}\left(f\right) = g\colon X \to Z \Rightarrow Y.$$

These constructions are mutually inverse, so $\mathsf{uncurry}\left(\mathsf{curry}\left(f\right)\right) = f$ and $\mathsf{curry}\left(\mathsf{uncurry}\left(g\right)\right) = g$. It also requires such an isomorphism to be natural in X and Y, which explicitly means that such that for all morphisms $g\colon U \to X$, the following commutes



(Nat-X)

and for all morphisms $h\colon Y \to V$, the following commutes



(Nat-Y)

Theorem A.4. *The two definitions of Cartesian closed category are equivalent.*

Proof. It suffices to show that the conditions (curry-$\exists$) and (curry-!) together are equivalent to the adjunction; we do this in two directions.

For the forward direction, we previously justified the existence of curry and uncurry constructions (as equations), and showed that they are mutually inverse, so it remains to show that the naturality conditions hold. For (Nat-X):

$$
\begin{aligned}
& \mathsf{curry}\left(f \circ g \times \mathrm{id}_Y\right) \\
= \quad & \{\ \text{universal property of exponential}\ \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{curry}\left(\mathsf{uncurry}\left(\mathsf{curry}\left(f\right)\right) \circ g \times \mathrm{id}_Y\right) \\
= \quad & \{\ \mathsf{uncurry}\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{Y,Z} \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_Y \circ g \times \mathrm{id}_Y\right) \\
= \quad & \{\ -\times Y \text{ functor}\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{Y,Z} \circ (\mathsf{curry}\left(f\right) \circ g) \times \mathrm{id}_Y\right) \\
= \quad & \{\ \mathsf{uncurry}\ \} \\
& \mathsf{curry}\left(\mathsf{uncurry}\left(\mathsf{curry}\left(f\right) \circ g\right)\right) \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& \mathsf{curry}\left(f\right) \circ g.
\end{aligned}
$$

Secondly, for (Nat-Y):

$$
\begin{aligned}
& \mathsf{curry}\left(h \circ f\right) \\
= \quad & \{\ \text{Lemma A.3}\ \} \\
& \mathrm{id}_Z \Rightarrow (h \circ f) \circ \mathsf{curry}\left(\mathrm{id}_{X\times Z}\right) \\
= \quad & \{\ Z \Rightarrow - \text{ functor}\ \} \\
& \mathrm{id}_Z \Rightarrow h \circ \mathrm{id}_Z \Rightarrow f \circ \mathsf{curry}\left(\mathrm{id}_{X\times Z}\right) \\
= \quad & \{\ \text{Lemma A.3}\ \} \\
& \mathrm{id}_Z \Rightarrow h \circ \mathsf{curry}\left(f\right).
\end{aligned}
$$

For the reverse direction, we assume the adjunction and derive (curry-$\exists$) and (curry-!). Examining (Nat-X), $\mathsf{curry}\left(f \circ g \times \mathrm{id}_Z\right) = \mathsf{curry}\left(f\right) \circ g$, so we can take the 'uncurrying' to derive $f \circ g \times \mathrm{id}_Z = \mathsf{uncurry}\left(\mathsf{curry}\left(f\right) \circ g\right)$.

$$
\begin{aligned}
& \mathsf{ev}_{X,Y} \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(\mathsf{ev}_{X,Y}\right)\right) \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{\ \text{uncurrying of (Nat-X)}\ \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(\mathrm{id}_Y \circ \mathsf{ev}_{X,Y} \circ \mathrm{id}_{X\Rightarrow Y\times X}\right)\right) \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{\ -\times X \text{ functorial}\ \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(\mathrm{id}_Y \circ \mathsf{ev}_{X,Y} \circ \mathrm{id}_{X\Rightarrow Y} \times \mathrm{id}_X\right)\right) \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{\ \text{action of } X \Rightarrow - \text{ on } \mathrm{id}_Y\ \} \\
& \mathsf{uncurry}\left(\mathrm{id}_X \Rightarrow \mathrm{id}_Y\right) \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{\ X \Rightarrow Y \text{ functorial}\ \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{uncurry}\left(\mathrm{id}_{X\Rightarrow Y}\right) \circ \mathsf{curry}\left(f\right) \times \mathrm{id}_X \\
= \quad & \{ \text{ uncurrying of (Nat-X) } \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(\mathsf{uncurry}\left(\mathrm{id}_{X\Rightarrow Y}\right)\right) \circ \mathsf{curry}\left(f\right)\right) \\
= \quad & \{ \text{ universal property of exponential } \} \\
& \mathsf{uncurry}\left(\mathrm{id}_{X\Rightarrow Y} \circ \mathsf{curry}\left(f\right)\right) \\
= \quad & \{ \text{ identities cancel through composition } \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(f\right)\right) \\
= \quad & \{ \text{ universal property of exponential } \} \\
& f.
\end{aligned}
$$

The proof for (curry-!) follows:

$$
\begin{aligned}
& \mathsf{curry}\left(\mathsf{ev}_{X,Y} \circ g \times \mathrm{id}_X\right) \\
= \quad & \{ \text{ (Nat-X) } \} \\
& \mathsf{curry}\left(\mathsf{ev}_{X,Y}\right) \circ g \\
= \quad & \{ \text{ identity cancels through composition } \} \\
& \mathsf{curry}\left(\mathrm{id}_Y \circ \mathsf{ev}_{X,Y} \circ \mathrm{id}_{X\Rightarrow Y\times X}\right) \circ g \\
= \quad & \{ \; - \times X \text{ functorial } \} \\
& \mathsf{curry}\left(\mathrm{id}_Y \circ \mathsf{ev}_{X,Y} \circ \mathrm{id}_{X\Rightarrow Y} \times \mathrm{id}_X\right) \circ g \\
= \quad & \{ \text{ action of } X \Rightarrow - \text{ functor on } \mathrm{id}_Y \; \} \\
& \mathrm{id}_X \Rightarrow \mathrm{id}_Y \circ g \\
= \quad & \{ \; X \Rightarrow - \text{ functorial } \} \\
& \mathrm{id}_{X\Rightarrow Y} \circ g \\
= \quad & \{ \text{ identity cancels through composition } \} \\
& g.
\end{aligned}
$$

□

It is also clear to see that (Nat-Y) corresponds to Lemma A.3.

This justifies that equational Cartesian closed categories and adjunctional Cartesian closed categories are one and the same. Specifically concerning the λ-calculus, the equational definition is much easier to work with as equations translate more directly into λ-terms than universal properties.

B PROOFS

Proposition 4.12. *The definition of model isomorphism suffices to define an isomorphism between two models.*

Proof. To prove this, we must illustrate that the composition of a model isomorphism and its inverse in either direction is equal to the identity morphism. Given $h\colon \mathbb{M} \to \mathbb{N}$, we construct its inverse $h^{-1}\colon \mathbb{N} \to \mathbb{M}$ by taking the inverse of each base component of h as base components:

$$h_X^{-1}(X) = \begin{cases} h_G^{-1}(\llbracket G \rrbracket_{\mathbb{N}}) & \text{if } X = \llbracket G \rrbracket_{\mathbb{N}}, G \in \mathsf{TV}, \\ (h_A^{-1} \times h_B^{-1})(\llbracket A \rrbracket_{\mathbb{N}} \times \llbracket B \rrbracket_{\mathbb{N}}) & \text{if } X = \llbracket A \times B \rrbracket_{\mathbb{N}}, A \times B \in \mathsf{ST}(\mathsf{TV}), \\ (h_A \Rightarrow h_B^{-1})(\llbracket A \rrbracket_{\mathbb{N}} \Rightarrow \llbracket B \rrbracket_{\mathbb{N}}) & \text{if } X = \llbracket A \to B \rrbracket_{\mathbb{N}}, A \to B \in \mathsf{ST}(\mathsf{TV}). \end{cases}$$

Now, we proceed by induction on the simple types of $\mathcal{T}$:

$G \in \mathsf{TV}$ **case** Immediate from the isomorphism of the components of h at ground types. ◁

$A \times B \in \mathsf{ST}(\mathsf{TV})$ **case**

$$\begin{aligned} & (h_A^{-1} \times h_B^{-1}) \circ (h_A \times h_B) \\ = \quad & \{ \text{ bifunctoriality of } \times \ \} \\ & (h_A^{-1} \circ h_A) \times (h_B^{-1} \circ h_B) \\ = \quad & \{ \text{ inductive hypothesis } \} \\ & \mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}}} \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}}} \\ = \quad & \{ \text{ bifunctoriality of } \times \ \} \\ & \mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}} \times \llbracket B \rrbracket_{\mathbb{M}}}. \end{aligned}$$

The reverse direction is exactly similar. ◁

$A \to B \in \mathsf{ST}(\mathsf{TV})$ **case**

$$\begin{aligned} & (h_A \Rightarrow h_B^{-1}) \circ (h_A^{-1} \Rightarrow h_B) \\ = \quad & \{ \ - \Rightarrow - \text{ contravariant in first argument, covariant in second } \} \\ & (h_A^{-1} \circ h_A \Rightarrow h_B^{-1} \circ h_B) \\ = \quad & \{ \text{ inductive hypothesis } \} \\ & (\mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}}} \Rightarrow \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}}}) \\ = \quad & \{ \text{ definition of } \Rightarrow \ \} \\ & \mathsf{curry}\left(\mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}}} \circ \mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}} \circ \mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}} \Rightarrow \llbracket B \rrbracket_{\mathbb{M}}} \times \mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}}}\right) \\ = \quad & \{ \text{ definition of } \mathsf{uncurry} \ \} \\ & \mathsf{curry}\left(\mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}}} \circ \mathsf{uncurry}\left(\mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}} \Rightarrow \llbracket B \rrbracket_{\mathbb{M}}}\right)\right) \\ = \quad & \{ \text{ id composition } \} \\ & \mathsf{curry}\left(\mathsf{uncurry}\left(\mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}} \Rightarrow \llbracket B \rrbracket_{\mathbb{M}}}\right)\right) \\ = \quad & \{ \text{ universal property } \} \\ & \mathrm{id}_{\llbracket A \rrbracket_{\mathbb{M}} \Rightarrow \llbracket B \rrbracket_{\mathbb{M}}} \end{aligned}$$

and similarly for the reverse direction. ◁

□

Proposition 4.15. *The interpretation of a type in $F_*(\mathbb{M})$ is isomorphic to the image of its interpretation in $\mathbb{M}$ in F* (ModTrans-Type).

Proof. We construct explicit witness to the canonical isomorphism $[\![A]\!]_{F_*(\mathbb{M})} \cong F([\![A]\!]_{\mathbb{M}})$ by structural induction over A:

$A = G \in \mathsf{TV}$ **case** Immediate from (ModTrans-Type-Base). ◁

$A = B \times C \in \mathsf{ST}(\mathsf{TV})$ **case**

$$
\begin{aligned}
& [\![B \times C]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Type-}\times) \ \} \\
& [\![B]\!]_{F_*(\mathbb{M})} \times [\![C]\!]_{F_*(\mathbb{M})} \\
\cong \quad & \{ \text{ inductive hypothesis } \} \\
& F([\![B]\!]_{\mathbb{M}}) \times F([\![C]\!]_{\mathbb{M}}) \\
\cong \quad & \{ \ F \text{ preserves finite products along } \Phi^{-1}_{[\![B]\!]_{\mathbb{M}} \times [\![C]\!]_{\mathbb{M}}} \ \} \\
& F([\![B]\!]_{\mathbb{M}} \times [\![C]\!]_{\mathbb{M}}) \\
= \quad & \{ \text{ (CatSem-}\times\text{), as } \mathbb{M} \text{ models } \mathcal{T} \ \} \\
& F([\![B \times C]\!]_{\mathbb{M}}).
\end{aligned}
$$

◁

$A = B \to C \in \mathsf{ST}(\mathsf{TV})$ **case**

$$
\begin{aligned}
& [\![B \to C]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Type-}\to\text{) } \} \\
& [\![B]\!]_{F_*(\mathbb{M})} \Rightarrow [\![C]\!]_{F_*(\mathbb{M})} \\
\cong \quad & \{ \text{ inductive hypothesis } \} \\
& F([\![B]\!]_{\mathbb{M}}) \Rightarrow F([\![C]\!]_{\mathbb{M}}) \\
\cong \quad & \{ \ F \text{ preserves exponentials along } \Psi^{-1}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![C]\!]_{\mathbb{M}}} \ \} \\
& F([\![B]\!]_{\mathbb{M}} \Rightarrow [\![C]\!]_{\mathbb{M}}) \\
= \quad & \{ \text{ (CatSem-}\to\text{), as } \mathbb{M} \text{ models } \mathcal{T} \ \} \\
& F(\{[\![B \to C]\!]_{\mathbb{M}}\}).
\end{aligned}
$$

◁

□

Proposition 4.16. *Given a model $\mathbb{M}$ of $\mathcal{T} = (\sigma, \mathcal{A})$ in $\mathscr{C}$, $F_*(\mathbb{M})$ is a well-defined model of $\mathcal{T}$ in $\mathscr{D}$.*

Proof. For a model to be well-defined, we would like $F_*(\mathbb{M})$ to satisfy Definition 3.1.

Every simple type $A \in \mathsf{ST}(\mathsf{TV})$ is interpreted by $F_*(\mathbb{M})$ by the canonical isomorphism given by (ModTrans-Type).

The proof that the interpretation of typing contexts is well-defined is essentially the same as the proof of the product case.

Turning our attention to terms-in-context $\Gamma \vdash x : B$ of $\mathcal{T}$, where $[\![\Gamma]\!]_{F_*(\mathbb{M})} = [\![A_1]\!]_{F_*(\mathbb{M})} \times \cdots \times [\![A_n]\!]_{F_*(\mathbb{M})}$, we proceed by induction over the structure of $[\![\Gamma \vdash x : B]\!]_{\mathbb{M}}$:

var case This implies that $B = A_i$ for some $1 \leq i \leq n$, and so

$$
\begin{aligned}
& [\![\Gamma + [x{:}B] + \Gamma' \vdash x : B]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& P_B^{-1} \circ F([\![\Gamma + [x{:}B] + \Gamma' \vdash x : B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ var rule, as } \mathbb{M} \text{ models } \mathcal{T} \text{ } \} \\
& P_B^{-1} \circ F(\pi_{i_{\mathscr{C}}} \colon [\![\Gamma]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}} \times [\![\Gamma']\!]_{\mathbb{M}} \to [\![B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ } F \text{ is a functor } \} \\
& P_B^{-1} \circ (F(\pi_{i_{\mathscr{C}}}) \colon F([\![\Gamma]\!]_{\mathbb{M}} \times [\![B]\!]_{\mathbb{M}} \times [\![\Gamma']\!]_{\mathbb{M}}) \to F([\![B]\!]_{\mathbb{M}})) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of product } \} \\
& P_B^{-1} \circ (\pi_{i_{\mathscr{D}}} \colon F([\![\Gamma]\!]_{\mathbb{M}}) \times F([\![B]\!]_{\mathbb{M}}) \times F([\![\Gamma']\!]_{\mathbb{M}}) \to F([\![B]\!]_{\mathbb{M}})) \circ \Phi_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \\
& \quad \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ inverses cancel } \} \\
& P_B^{-1} \circ (\pi_{i_{\mathscr{D}}} \colon F([\![\Gamma]\!]_{\mathbb{M}}) \times F([\![B]\!]_{\mathbb{M}}) \times F([\![\Gamma']\!]_{\mathbb{M}}) \to F([\![B]\!]_{\mathbb{M}})) \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of product } \} \\
& P_B^{-1} \circ P_B \circ \pi_{i_{\mathscr{D}}} \colon [\![\Gamma]\!]_{F_*(\mathbb{M})} \times [\![B]\!]_{F_*(\mathbb{M})} \times [\![\Gamma']\!]_{F_*(\mathbb{M})} \to [\![B]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ inverses cancel } \} \\
& \pi_{i_{\mathscr{D}}} \colon [\![\Gamma]\!]_{F_*(\mathbb{M})} \times [\![B]\!]_{F_*(\mathbb{M})} \times [\![\Gamma']\!]_{F_*(\mathbb{M})} \to [\![B]\!]_{F_*(\mathbb{M})}.
\end{aligned}
$$

◁

unit case

$$
\begin{aligned}
& [\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& P^{-1}_{\mathsf{unit}} \circ F([\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \colon [\![\Gamma]\!]_{F_*(\mathbb{M})} \to [\![\mathsf{unit}]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Type-unit) } \} \\
& P^{-1}_{\mathsf{unit}} \circ F([\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \colon [\![\Gamma]\!]_{F_*(\mathbb{M})} \to 1_{\mathscr{D}} \\
= \quad & \{ \text{ universal property of terminal object } \} \\
& !_{\mathscr{D}} \colon [\![\Gamma]\!]_{F_*(\mathbb{M})} \to 1_{\mathscr{D}}.
\end{aligned}
$$

◁

const case

$$
\begin{aligned}
& [\![\Gamma \vdash c : A]\!]_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \}
\end{aligned}
$$

$$
\begin{aligned}
& P_A^{-1} \circ F(\llbracket \Gamma \vdash c : A \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ const rule, as } \mathbb{M} \text{ models } \mathcal{T} \text{ } \} \\
& P_A^{-1} \circ F(\llbracket c \rrbracket_{\mathbb{M}} \circ !_{\mathscr{C}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ } F \text{ is a functor } \} \\
& P_A^{-1} \circ F(\llbracket c \rrbracket_{\mathbb{M}}) \circ F(!_{\mathscr{C}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of terminal object } \} \\
& P_A^{-1} \circ F(\llbracket c \rrbracket_{\mathbb{M}}) \circ \Phi^{-1} \circ !_{\mathscr{D}} \\
= \quad & \{ \text{ } P_{\mathsf{unit}} = \mathrm{id}_{1_{\mathscr{D}}}, \text{(ModTrans-FSym) } \} \\
& \llbracket c \rrbracket_{F_*(\mathbb{M})} \circ !_{\mathscr{D}}.
\end{aligned}
$$

◁

func case

$$
\begin{aligned}
& \llbracket \Gamma \vdash f(t_1, \ldots, t_n) : B \rrbracket_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& P_B^{-1} \circ F(\llbracket \Gamma \vdash f(t_1, \ldots, t_n) : B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ func rule, as } \mathbb{M} \text{ models } \mathcal{T} \text{ } \} \\
& P_B^{-1} \circ F(\llbracket f \rrbracket_{\mathbb{M}} \circ \langle \llbracket t_1 \rrbracket_{\mathbb{M}}, \ldots, \llbracket t_n \rrbracket_{\mathbb{M}} \rangle) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ } F \text{ is a functor } \} \\
& P_B^{-1} \circ F(\llbracket f \rrbracket_{\mathbb{M}}) \circ F(\langle \llbracket t_1 \rrbracket_{\mathbb{M}}, \ldots, \llbracket t_n \rrbracket_{\mathbb{M}} \rangle) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of product } \} \\
& P_B^{-1} \circ F(\llbracket f \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ \langle F(\llbracket t_1 \rrbracket_{\mathbb{M}}), \ldots, F(\llbracket t_n \rrbracket_{\mathbb{M}}) \rangle \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ inverses cancel } \} \\
& P_B^{-1} \circ F(\llbracket f \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
& \quad \circ P_{A_1}^{-1} \times \cdots \times P_{A_n}^{-1} \circ \langle F(\llbracket t_1 \rrbracket_{\mathbb{M}}), \ldots, F(\llbracket t_n \rrbracket_{\mathbb{M}}) \rangle \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ (ModTrans-FSym) } \} \\
& \llbracket f \rrbracket_{F_*(\mathbb{M})} \circ P_{A_1}^{-1} \times \cdots \times P_{A_n}^{-1} \circ \langle F(\llbracket t_1 \rrbracket_{\mathbb{M}}), \ldots, F(\llbracket t_n \rrbracket_{\mathbb{M}}) \rangle \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ composition over products } \} \\
& \llbracket f \rrbracket_{F_*(\mathbb{M})} \circ \langle P_{A_1}^{-1} \circ F(\llbracket t_1 \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}, \\
& \quad \ldots, P_{A_n}^{-1} \circ F(\llbracket t_n \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \rangle \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& \llbracket f \rrbracket_{F_*(\mathbb{M})} \circ \langle \llbracket t_1 \rrbracket_{F_*(\mathbb{M})}, \ldots, \llbracket t_n \rrbracket_{F_*(\mathbb{M})} \rangle.
\end{aligned}
$$

◁

abs case

$$
\begin{aligned}
& \llbracket \Gamma \vdash (\lambda x{:}A.t) : A \to B \rrbracket_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& P^{-1}_{A\to B} \circ F(\llbracket \Gamma \vdash (\lambda x{:}A.t) : A \to B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ abs rule, as } \mathbb{M} \text{ models } \mathcal{T} \} \\
& P^{-1}_{A\to B} \circ F(\mathsf{curry}\left(\llbracket \Gamma + [x{:}A] \vdash t : B \rrbracket_{\mathbb{M}}\right)) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of exponential } \} \\
& \mathsf{curry}\left(P^{-1}_{B} \circ F(\llbracket \Gamma + [x{:}A] \vdash t : B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} \times \llbracket A \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \times P_A\right) \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& \mathsf{curry}\left(\llbracket \Gamma + [x{:}A] \vdash t : B \rrbracket_{F_*(\mathbb{M})}\right).
\end{aligned}
$$

◁

app case

$$
\begin{aligned}
& \llbracket \Gamma \vdash uv : B \rrbracket_{F_*(\mathbb{M})} \\
= \quad & \{ \text{ (ModTrans-Term) } \} \\
& P^{-1}_{B} \circ F(\llbracket \Gamma \vdash uv : B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ app rule, as } \mathbb{M} \text{ models } \mathcal{T} \} \\
& P^{-1}_{B} \circ F(\mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}} \circ \langle \llbracket \Gamma \vdash u : A \to B \rrbracket_{\mathbb{M}}, \llbracket \Gamma \vdash v : A \rrbracket_{\mathbb{M}} \rangle) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ } F \text{ is a functor } \} \\
& P^{-1}_{B} \circ F(\mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}}) \circ F(\langle \llbracket \Gamma \vdash u : A \to B \rrbracket_{\mathbb{M}}, \llbracket \Gamma \vdash v : A \rrbracket_{\mathbb{M}} \rangle) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ universal property of product } \} \\
& P^{-1}_{B} \circ F(\mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}}) \circ \Phi^{-1}_{\llbracket A \to B \rrbracket_{\mathbb{M}} \times \llbracket A \rrbracket_{\mathbb{M}}} \\
& \quad \circ \langle F(\llbracket \Gamma \vdash u : A \to B \rrbracket_{\mathbb{M}}), F(\llbracket \Gamma \vdash v : A \rrbracket_{\mathbb{M}}) \rangle \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ inverses cancel } \} \\
& P^{-1}_{B} \circ F(\mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}}) \circ \Phi^{-1}_{\llbracket A \to B \rrbracket_{\mathbb{M}} \times \llbracket A \rrbracket_{\mathbb{M}}} \circ P_{A\to B} \times P_A \\
& \quad \circ P^{-1}_{A\to B} \times P^{-1}_{A} \circ \langle F(\llbracket \Gamma \vdash u : A \to B \rrbracket_{\mathbb{M}}), F(\llbracket \Gamma \vdash v : A \rrbracket_{\mathbb{M}}) \rangle \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad & \{ \text{ composition over product } \} \\
& P^{-1}_{B} \circ F(\mathsf{ev}_{\llbracket A \rrbracket_{\mathbb{M}}, \llbracket B \rrbracket_{\mathbb{M}}}) \circ \Phi^{-1}_{\llbracket A \to B \rrbracket_{\mathbb{M}} \times \llbracket A \rrbracket_{\mathbb{M}}} \circ P_{A\to B} \times P_A \\
& \quad \circ \langle P^{-1}_{A\to B} \circ F(\llbracket \Gamma \vdash u : A \to B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}, \\
& \quad P^{-1}_{A} \circ F(\llbracket \Gamma \vdash v : A \rrbracket_{\mathbb{M}} \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}) \rangle \\
= \quad & \{ \text{ (ModTrans-Term) } \}
\end{aligned}
$$

$P_B^{-1} \circ F(\mathsf{ev}_{[\![A]\!]_{\mathbb{M}},[\![B]\!]_{\mathbb{M}}}) \circ \Phi^{-1}_{[\![A\to B]\!]_{\mathbb{M}}\times[\![A]\!]_{\mathbb{M}}} \circ P_{A\to B} \times P_A \circ \langle [\![\Gamma \vdash u : A \to B]\!]_{F_*(\mathbb{M})}, [\![\Gamma \vdash v : A]\!]_{F_*(\mathbb{M})}\rangle$

= { inductive hypothesis: $[\![A \to B]\!]_{F_*(\mathbb{M})} = [\![A]\!]_{F_*(\mathbb{M})} \Rightarrow [\![B]\!]_{F_*(\mathbb{M})}$, universal property of exponential }

$\mathsf{ev}_{[\![A]\!]_{F_*(\mathbb{M})},[\![B]\!]_{F_*(\mathbb{M})}} \circ \langle [\![\Gamma \vdash u : A \to B]\!]_{F_*(\mathbb{M})}, [\![\Gamma \vdash v : A]\!]_{F_*(\mathbb{M})}\rangle$.

◁

pair case

$[\![\Gamma \vdash \langle u, v\rangle : A \times B]\!]_{F_*(\mathbb{M})}$

= { (ModTrans-Term) }

$P^{-1}_{A\times B} \circ F([\![\Gamma \vdash \langle u, v\rangle : A \times B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { pair rule, as $\mathbb{M}$ models $\mathcal{T}$ }

$P^{-1}_{A\times B} \circ F(\langle [\![\Gamma \vdash u : A]\!]_{\mathbb{M}}, [\![\Gamma \vdash v : B]\!]_{\mathbb{M}}\rangle) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { universal property of product }

$P^{-1}_{A\times B} \circ \Phi^{-1}_{[\![A]\!]_{\mathbb{M}}\times[\![B]\!]_{\mathbb{M}}} \circ \langle F([\![\Gamma \vdash u : A]\!]_{\mathbb{M}}), F([\![\Gamma \vdash v : B]\!]_{\mathbb{M}})\rangle \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { inductive hypothesis: $[\![A \times B]\!]_{F_*(\mathbb{M})} = [\![A \times B]\!]_{F_*(\mathbb{M})}$, $P^{-1}_{A\times B} = P_A^{-1} \times P_B^{-1} \circ \Phi_{[\![A]\!]_{\mathbb{M}}\times[\![B]\!]_{\mathbb{M}}}$ by canonicity }

$P_A^{-1} \times P_B^{-1} \circ \Phi_{[\![A]\!]_{\mathbb{M}}\times[\![B]\!]_{\mathbb{M}}} \circ \Phi^{-1}_{[\![A]\!]_{\mathbb{M}}\times[\![B]\!]_{\mathbb{M}}} \circ \langle F([\![\Gamma \vdash u : A]\!]_{\mathbb{M}}), F([\![\Gamma \vdash v : B]\!]_{\mathbb{M}})\rangle$
$\circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { inverses cancel }

$P_A^{-1} \times P_B^{-1} \circ \langle F([\![\Gamma \vdash u : A]\!]_{\mathbb{M}}), F([\![\Gamma \vdash v : B]\!]_{\mathbb{M}})\rangle \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { composition over product }

$\langle P_A^{-1} \circ F([\![\Gamma \vdash u : A]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n},$
$P_B^{-1} \circ F([\![\Gamma \vdash v : B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}\rangle$

= { (ModTrans-Term) }

$\langle [\![\Gamma \vdash u : A]\!]_{F_*(\mathbb{M})}, [\![\Gamma \vdash v : B]\!]_{F_*(\mathbb{M})}\rangle$.

◁

fst, snd case

$[\![\Gamma \vdash \mathsf{fst}\,(t) : A]\!]_{F_*(\mathbb{M})}$

= { (ModTrans-Term) }

$P_A^{-1} \circ F([\![\Gamma \vdash \mathsf{fst}\,(t) : A]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { fst rule, as $\mathbb{M}$ models $\mathcal{T}$ }

$P_A^{-1} \circ F(\pi_{1_{\mathscr{C}}} \circ [\![\Gamma \vdash t : A \times B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { F is a functor }

$P_A^{-1} \circ F(\pi_{1_{\mathscr{C}}}) \circ F([\![\Gamma \vdash t : A \times B]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}}\times\cdots\times[\![A_n]\!]_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n}$

= { inductive hypothesis: $[\![A \times B]\!]_{F_*(\mathbb{M})} = [\![A \times B]\!]_{F_*(\mathbb{M})}$, universal property of product }

$$\begin{aligned}
&\pi_{1_{\mathscr{D}}} \circ P^{-1}_{A\times B} \circ F(\llbracket \Gamma \vdash t : A \times B \rrbracket_{\mathbb{M}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ P_{A_1} \times \cdots \times P_{A_n} \\
= \quad &\{ \text{ (ModTrans-Term) } \} \\
&\pi_{1_{\mathscr{D}}} \circ \llbracket \Gamma \vdash t : A \times B \rrbracket_{F_*(\mathbb{M})}.
\end{aligned}$$

The case for **snd** is similar. ◁

Therefore, we deduce that the collection of objects and morphisms obtained in $\mathscr{D}$, $F_*(\mathbb{M})$, is a model of $\mathcal{T}$ by the Soundness Theorem, as it is a structure of σ in $\mathscr{D}$.

□

Proposition 4.19. *Given a model isomorphism* $h \colon \mathbb{M} \xrightarrow{\sim} \mathbb{N}$*, for all* $A \in \mathsf{ST}(\mathsf{TV})$*,*

$$F_*(h)_A \colon \llbracket A \rrbracket_{F_*(\mathbb{M})} \xrightarrow{\sim} \llbracket A \rrbracket_{F_*(\mathbb{N})} = P^{-1}_{A\,\mathbb{N}} \circ F(h_A) \circ P_{A\,\mathbb{M}}. \tag{ModTrans-MIso}$$

Proof. By induction over A.

$A = G \in \mathsf{TV}$ **case** Immediate from (ModTrans-MIso-Base) (recall that P_G is an identity morphism). ◁

$A = B \times C \in \mathsf{ST}(\mathsf{TV})$ **case**

$$\begin{aligned}
&F_*(h)_{B\times C} \\
= \quad &\{ \text{ (MIso-Components) } \} \\
&F_*(h)_B \times F_*(h)_C \\
= \quad &\{ \text{ inductive hypothesis } \} \\
&P^{-1}_{B\,\mathbb{N}} \circ F(h_B) \circ P_{B\,\mathbb{M}} \times P^{-1}_{C\,\mathbb{N}} \circ F(h_C) \circ P_{C\,\mathbb{M}} \\
= \quad &\{ \text{ bifunctoriality of } - \times - \} \\
&P^{-1}_{B\,\mathbb{N}} \times P^{-1}_{C\,\mathbb{N}} \circ F(h_B) \times F(h_C) \circ P_{B\,\mathbb{M}} \times P_{C\,\mathbb{M}} \\
= \quad &\{ \text{ Lemma 4.4 } \} \\
&P^{-1}_{B\,\mathbb{N}} \times P^{-1}_{C\,\mathbb{N}} \circ \Phi_{\llbracket B \rrbracket_{\mathbb{N}} \times \llbracket C \rrbracket_{\mathbb{N}}} \circ F(h_B \times h_C) \circ \Phi^{-1}_{\llbracket B \rrbracket_{\mathbb{M}} \times \llbracket C \rrbracket_{\mathbb{M}}} \circ P_{B\,\mathbb{M}} \times P_{C\,\mathbb{M}} \\
= \quad &\{ \text{ Lemma 4.17 } \} \\
&P^{-1}_{B\times C\,\mathbb{N}} \circ F(h_B \times h_C) \circ P_{B\times C\,\mathbb{M}} \\
= \quad &\{ \text{ (MIso-Components) } \} \\
&P^{-1}_{B\times C\,\mathbb{N}} \circ F(h_{B\times C}) \circ P_{B\times C\,\mathbb{M}}.
\end{aligned}$$

◁

$A = B \to C \in \mathsf{ST}(\mathsf{TV})$ **case**

$$\begin{aligned}
&F_*(h)_{B\Rightarrow C} \\
= \quad &\{ \text{ (MIso-Components) } \} \\
&(F_*(h)_B)^{-1} \Rightarrow F_*(h)_C \\
= \quad &\{ \text{ inductive hypothesis } \} \\
&\left(P^{-1}_{B\,\mathbb{N}} \circ F(h_B) \circ P_{B\,\mathbb{M}}\right)^{-1} \Rightarrow P^{-1}_{C\,\mathbb{N}} \circ F(h_C) \circ P_{C\,\mathbb{M}} \\
= \quad &\{ \left(P^{-1}_{B\,\mathbb{N}} \circ F(h_B) \circ P_{B\,\mathbb{M}}\right)^{-1} = P^{-1}_{B\,\mathbb{M}} \circ F(h_B)^{-1} \circ P_{B\,\mathbb{N}} \ \}
\end{aligned}$$

$$
\begin{aligned}
& {P_B^{-1}}_{\mathbb{M}} \circ F(h_B)^{-1} \circ {P_B}_{\mathbb{N}} \Rightarrow {P_C^{-1}}_{\mathbb{N}} \circ F(h_C) \circ {P_C}_{\mathbb{M}} \\
= \quad & \{\ F \text{ is a functor, so it preserve isomorphisms } \} \\
& {P_B^{-1}}_{\mathbb{M}} \circ F(h_B^{-1}) \circ {P_B}_{\mathbb{N}} \Rightarrow {P_C^{-1}}_{\mathbb{N}} \circ F(h_C) \circ {P_C}_{\mathbb{M}} \\
= \quad & \{\ \text{bifunctoriality of } - \Rightarrow -, \text{ contravariant in first argument } \} \\
& {P_B}_{\mathbb{N}} \Rightarrow {P_C^{-1}}_{\mathbb{N}} \circ F(h_B^{-1}) \Rightarrow F(h_C) \circ {P_B^{-1}}_{\mathbb{M}} \Rightarrow {P_C}_{\mathbb{M}} \\
= \quad & \{\ \text{Lemma 4.6 } \} \\
& {P_B}_{\mathbb{N}} \Rightarrow {P_C^{-1}}_{\mathbb{N}} \circ \Psi_{[\![B]\!]_{\mathbb{N}} \Rightarrow [\![C]\!]_{\mathbb{N}}} \circ F(h_B^{-1} \Rightarrow h_C) \circ \Psi^{-1}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![C]\!]_{\mathbb{M}}} \circ {P_B^{-1}}_{\mathbb{M}} \Rightarrow {P_C}_{\mathbb{M}} \\
= \quad & \{\ \text{Lemma 4.18 } \} \\
& {P^{-1}_{B \to C}}_{\mathbb{N}} \circ F(h_B^{-1} \Rightarrow h_C) \circ {P_{B \to C}}_{\mathbb{M}} \\
= \quad & \{\ \text{(MIso-Components) } \} \\
& {P^{-1}_{B \to C}}_{\mathbb{N}} \circ F(h_{B \Rightarrow C}) \circ {P_{B \to C}}_{\mathbb{M}}.
\end{aligned}
$$

◁

□

Proposition 4.21. *Given a model isomorphism $h\colon \mathbb{M} \to \mathbb{N}$, $F_*(h)\colon F_*(\mathbb{M}) \to F_*(\mathbb{N})$ is a model isomorphism.*

Proof. First, we establish that $F_*(h)$ is a valid model homomorphism, satisfying Definition 4.10. The first commutativity condition we must fulfil is

$$
\begin{array}{ccc}
[\![A_1]\!]_{F_*(\mathbb{M})} \times \cdots \times [\![A_n]\!]_{F_*(\mathbb{M})} & \xrightarrow{[\![f]\!]_{F_*(\mathbb{M})}} & [\![B]\!]_{F_*(\mathbb{M})} \\
\downarrow{\scriptstyle F_*(h)_{A_1} \times \cdots \times F_*(h)_{A_n}} & & \downarrow{\scriptstyle F_*(h)_B} \\
[\![A_1]\!]_{F_*(\mathbb{N})} \times \cdots \times [\![A_n]\!]_{F_*(\mathbb{N})} & \xrightarrow[[\![f]\!]_{F_*(\mathbb{N})}]{} & [\![B]\!]_{F_*(\mathbb{N})}
\end{array}
$$

$$
\begin{aligned}
& F_*(h)_B \circ [\![f]\!]_{F_*(\mathbb{M})} \\
= \quad & \{\ \text{(ModTrans-MIso), (ModTrans-FSym) } \} \\
& {P_B^{-1}}_{\mathbb{N}} \circ F(h_B) \circ {P_B}_{\mathbb{M}} \circ {P_B^{-1}}_{\mathbb{M}} \circ F([\![f]\!]_{\mathbb{M}}) \circ \Phi^{-1}_{[\![A_1]\!]_{\mathbb{M}} \times \cdots \times [\![A_n]\!]_{\mathbb{M}}} \circ {P_{A_1}}_{\mathbb{M}} \times \cdots \times {P_{A_n}}_{\mathbb{M}} \\
= \quad & \{\ \text{inverses cancel, Lemma 4.17 } \} \\
& {P_B^{-1}}_{\mathbb{N}} \circ F(h_B) \circ F([\![f]\!]_{\mathbb{M}}) \circ {P_{A_1 \times \cdots \times A_n}}_{\mathbb{M}} \\
= \quad & \{\ F \text{ is a functor } \} \\
& {P_B^{-1}}_{\mathbb{N}} \circ F(h_B \circ [\![f]\!]_{\mathbb{M}}) \circ {P_{A_1 \times \cdots \times A_n}}_{\mathbb{M}} \\
= \quad & \{\ h \text{ is a model homomorphism } \} \\
& {P_B^{-1}}_{\mathbb{N}} \circ F([\![f]\!]_{\mathbb{N}} \circ h_{A_1} \times \cdots \times h_{A_n}) \circ {P_{A_1 \times \cdots \times A_n}}_{\mathbb{M}} \\
= \quad & \{\ \text{(MIso-Components) } \}
\end{aligned}
$$

$$
\begin{aligned}
& P_B^{-1}{}_{\mathbb{N}} \circ F(\llbracket f \rrbracket_{\mathbb{N}} \circ h_{A_1 \times \cdots \times A_n}) \circ P_{A_1 \times \cdots \times A_n \mathbb{M}} \\
= \quad & \{\ F \text{ is a functor } \} \\
& P_B^{-1}{}_{\mathbb{N}} \circ F(\llbracket f \rrbracket_{\mathbb{N}}) \circ F(h_{A_1 \times \cdots \times A_n}) \circ P_{A_1 \times \cdots \times A_n \mathbb{M}} \\
= \quad & \{\ \text{inverses cancel } \} \\
& P_B^{-1}{}_{\mathbb{N}} \circ F(\llbracket f \rrbracket_{\mathbb{N}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{N}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{N}}} \circ \Phi_{\llbracket A_1 \rrbracket_{\mathbb{N}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{N}}} \circ F(h_{A_1 \times \cdots \times A_n}) \circ P_{A_1 \times \cdots \times A_n \mathbb{M}} \\
= \quad & \{\ \text{inverses cancel } \} \\
& P_B^{-1}{}_{\mathbb{N}} \circ F(\llbracket f \rrbracket_{\mathbb{N}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{N}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{N}}} \circ P_{A_1 \mathbb{N}} \times \cdots \times P_{A_n \mathbb{N}} \\
& \quad \circ P^{-1}_{A_1 \mathbb{N}} \times \cdots \times P^{-1}_{A_n \mathbb{N}} \circ \Phi_{\llbracket A_1 \rrbracket_{\mathbb{N}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{N}}} \circ F(h_{A_1 \times \cdots \times A_n}) \circ P_{A_1 \times \cdots \times A_n \mathbb{M}} \\
= \quad & \{\ \text{Lemma 4.17 } \} \\
& P_B^{-1}{}_{\mathbb{N}} \circ F(\llbracket f \rrbracket_{\mathbb{N}}) \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{N}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{N}}} \circ P_{A_1 \mathbb{N}} \times \cdots \times P_{A_n \mathbb{N}} \\
& \quad \circ P^{-1}_{A_1 \times \cdots \times A_n \mathbb{N}} \circ F(h_{A_1 \times \cdots \times A_n}) \circ P_{A_1 \times \cdots \times A_n \mathbb{M}} \\
= \quad & \{\ \text{(ModTrans-FSym), (ModTrans-MIso) } \} \\
& \llbracket f \rrbracket_{F_*(\mathbb{N})} \circ F_*(h)_{A_1 \times \cdots \times A_n}.
\end{aligned}
$$

The remaining proof cases proceed similarly.

This establishes that $F_*(h)$ is a valid model homomorphism; as the each base component $F_*(h)_G$ is $F(h_G)$, and as functors preserve isomorphisms, $F_*(h)$ is a model isomorphism by Proposition 4.12.

□

Proposition 4.23. *$\phi_* \mathbb{M}$ is a model isomorphism.*

Proof. By Proposition 4.16, $F_*(\mathbb{M})$ and $E_*(\mathbb{M})$ are both models of $\mathcal{T}$ in $\mathscr{D}$. We seek to fulfil

$$
\begin{array}{ccc}
\llbracket A_1 \rrbracket_{F_*(\mathbb{M})} \times \cdots \times \llbracket A_n \rrbracket_{F_*(\mathbb{M})} & \xrightarrow{\llbracket f \rrbracket_{F_*(\mathbb{M})}} & \llbracket B \rrbracket_{F_*(\mathbb{M})} \\
\downarrow{\scriptstyle \phi_* \mathbb{M}_{A_1} \times \cdots \times \phi_* \mathbb{M}_{A_n}} & & \downarrow{\scriptstyle \phi_* \mathbb{M}_B} \\
\llbracket A_1 \rrbracket_{E_*(\mathbb{M})} \times \cdots \times \llbracket A_n \rrbracket_{E_*(\mathbb{M})} & \xrightarrow[\llbracket f \rrbracket_{E_*(\mathbb{M})}]{} & \llbracket B \rrbracket_{E_*(\mathbb{M})}
\end{array}
$$

Expanding $\llbracket f \rrbracket_{F_*(\mathbb{M})}$ by (ModTrans-FSym) yields[14]

$$
\begin{array}{ccccc}
F(\llbracket A_1 \rrbracket_{\mathbb{M}}) \times \cdots \times F(\llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow{\Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}}} & F(\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow{F(\llbracket f \rrbracket_{\mathbb{M}})} & F(\llbracket B \rrbracket_{\mathbb{M}}) \\
\downarrow{\scriptstyle \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}}} \times \cdots \times \phi_{\llbracket A_n \rrbracket_{\mathbb{M}}}} & 1 & \vdots{\scriptstyle \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}}} & 2 & \downarrow{\scriptstyle \phi_{\llbracket B \rrbracket_{\mathbb{M}}}} \\
E(\llbracket A_1 \rrbracket_{\mathbb{M}}) \times \cdots \times E(\llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow[\Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}_E}}]{} & E(\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow[E(\llbracket f \rrbracket_{\mathbb{M}})]{} & E(\llbracket B \rrbracket_{\mathbb{M}})
\end{array}
$$

[14] The isomorphisms $\llbracket A_1 \rrbracket_{F_*(\mathbb{M})} \times \cdots \times \llbracket A_n \rrbracket_{F_*(\mathbb{M})} \xrightarrow{\sim} F(\llbracket A_1 \rrbracket_{\mathbb{M}}) \times \cdots \times F(\llbracket A_n \rrbracket_{\mathbb{M}})$, and $F(\llbracket B \rrbracket_{\mathbb{M}}) \xrightarrow{\sim} \llbracket B \rrbracket_{F_*(\mathbb{M})}$, and their correspondents for E have been elided.

The commutativity of 2 follows from the naturality of ϕ, so it suffices to show that 1 commutes. By naturality, we have the following for each $1 \leq i \leq n$

$$\begin{array}{ccc} F(\llbracket A_1 \rrbracket_{\mathbb{M}}) \times \cdots \times F(\llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow{F(\pi_i)} & F(\llbracket A_i \rrbracket_{\mathbb{M}}) \\ \downarrow{\scriptstyle \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}}} & & \downarrow{\scriptstyle \phi_{\llbracket A_i \rrbracket_{\mathbb{M}}}} \\ E(\llbracket A_1 \rrbracket_{\mathbb{M}}) \times \cdots \times E(\llbracket A_n \rrbracket_{\mathbb{M}}) & \xrightarrow[E(\pi_i)]{} & E(\llbracket A_i \rrbracket_{\mathbb{M}}) \end{array}$$

and so

$$\begin{array}{ll} & \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}}} \times \cdots \times \phi_{\llbracket A_n \rrbracket_{\mathbb{M}}} \circ \Phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} F} \\ = & \quad \{ \text{ (PP-Functor) } \} \\ & \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}}} \times \cdots \times \phi_{\llbracket A_n \rrbracket_{\mathbb{M}}} \circ \langle F(\pi_1), \ldots, F(\pi_n) \rangle \\ = & \quad \{ \text{ composition through products } \} \\ & \langle \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}}} \circ F(\pi_1), \ldots, \phi_{\llbracket A_n \rrbracket_{\mathbb{M}}} \circ F(\pi_n) \rangle \\ = & \quad \{ \text{ naturality of } \phi \text{ (above) } \} \\ & \langle E(\pi_1) \circ \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}}, E(\pi_n) \circ \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \rangle \\ = & \quad \{ \text{ composition through products } \} \\ & \langle E(\pi_1), \ldots, E(\pi_n) \rangle \circ \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \\ = & \quad \{ \text{ (PP-Functor) } \} \\ & \Phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} E} \circ \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \\ \Longrightarrow & \quad \{ \text{ pre-composition by } \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} E}, \text{ post-composition by } \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} F} \} \\ = & \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} E} \circ \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}}} \times \cdots \times \phi_{\llbracket A_n \rrbracket_{\mathbb{M}}} \\ & \phi_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}}} \circ \Phi^{-1}_{\llbracket A_1 \rrbracket_{\mathbb{M}} \times \cdots \times \llbracket A_n \rrbracket_{\mathbb{M}} F} \end{array}$$

as required. The other proof cases follow analogously.

As ϕ is a natural isomorphism, its components are all isomorphisms; in particular, the base components of $\phi_* \mathbb{M}$ are all isomorphisms, so by Proposition 4.12 it is a model isomorphism.

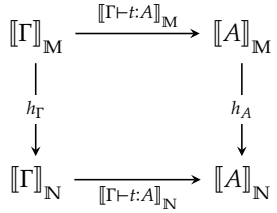
□

Proposition 4.29. *Given a model isomorphism $h \colon \mathbb{M} \to \mathbb{N}$, $\mathbf{Ap}^{-1}_{\mathbb{G}}(h)$ is a natural isomorphism.*

Proof. The naturality square we must fulfil is

$$\begin{array}{ccc} \mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{M})(A) = \llbracket A \rrbracket_{\mathbb{M}} & \xrightarrow{\mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{M})(\lceil t \rceil^{A \to B}]_{\mathcal{T}}) = \llbracket \vdash t : A \to B \rrbracket_{\mathbb{M}}} & \llbracket B \rrbracket_{\mathbb{M}} = \mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{M})(B) \\ \downarrow{\scriptstyle h_A} & & \downarrow{\scriptstyle h_B} \\ \mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{N})(A) = \llbracket A \rrbracket_{\mathbb{N}} & \xrightarrow[\mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{N})(\lceil t \rceil^{A \to B}]_{\mathcal{T}}) = \llbracket \vdash t : A \to B \rrbracket_{\mathbb{N}}}{} & \llbracket B \rrbracket_{\mathbb{N}} = \mathbf{Ap}^{-1}_{\mathbb{G}}(\mathbb{N})(B) \end{array}$$

Instead, we focus on the more general property



which will imply what is required.
We proceed by induction on the derivation of $\Gamma \vdash t : A$.

var case

$$
\begin{aligned}
& [\![\Gamma + [t{:}A] + \Gamma' \vdash t : A]\!]_{\mathbb{N}} \circ h_{\Gamma+[t:A]+\Gamma'} \\
= \quad & \{ \text{ var rule, (MIso-Components) } \} \\
& \pi_i \circ h_\Gamma \times h_A \times h_{\Gamma'} \\
= \quad & \{ \text{ action of } - \times - \text{ bifunctor on morphisms } \} \\
& \pi_i \circ \langle \dots , h_A \circ \pi_{i'} \dots \rangle \\
= \quad & \{ \text{ universal property of product } \} \\
& h_A \circ \pi_i \\
= \quad & \{ \text{ var rule } \} \\
& h_A \circ [\![\Gamma + [t{:}A] + \Gamma' \vdash t : A]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

unit case Then $t = \langle\rangle$ and $A = \mathsf{unit}$, so $h_{\mathsf{unit}} \colon \mathsf{unit} \to \mathsf{unit} = \mathrm{id}_{\mathsf{unit}}$ by the universal property of terminal object, and

$$
\begin{aligned}
& [\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ unit rule, (MIso-Components) } \} \\
& !_{[\![\Gamma]\!]_{\mathbb{N}}} \circ h_\Gamma \\
= \quad & \{ \text{ universal property of terminal object } \} \\
& !_{[\![\Gamma]\!]_{\mathbb{M}}} \\
= \quad & \{ \text{ identity cancels through composition } \} \\
& h_{\mathsf{unit}} \circ ! \\
= \quad & \{ \text{ unit rule } \} \\
& h_{\mathsf{unit}} \circ [\![\Gamma \vdash \langle\rangle : \mathsf{unit}]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

const case

$$
\begin{aligned}
& [\![\Gamma \vdash t : A]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ const rule } \} \\
& [\![t]\!]_{\mathbb{N}} \circ !_{[\![\Gamma]\!]_{\mathbb{N}}} \circ h_\Gamma \\
= \quad & \{ \text{ universal property of terminal object } \}
\end{aligned}
$$

$$
\begin{aligned}
& [\![t]\!]_{\mathbb{N}} \circ !_{[\![\Gamma]\!]_{\mathbb{M}}} \\
= \quad & \{ \text{ (MHom-Const) } \} \\
& h_A \circ [\![t]\!]_{\mathbb{M}} \circ !_{[\![\Gamma]\!]_{\mathbb{M}}} \\
= \quad & \{ \text{ const rule } \} \\
& h_A \circ [\![\Gamma \vdash t : A]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

func case Then $t = f(u_1, \ldots, u_n)$ for each $\Gamma \vdash u_i : B_i$, and

$$
\begin{aligned}
& [\![\Gamma \vdash f(u_1, \ldots, u_n) : A]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ func rule } \} \\
& [\![f]\!]_{\mathbb{N}} \circ \langle [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{N}}, \ldots, [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \rangle \circ h_\Gamma \\
= \quad & \{ \text{ composition distributes over product } \} \\
& [\![f]\!]_{\mathbb{N}} \circ \langle [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{N}} \circ h_\Gamma, \ldots, [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \circ h_\Gamma \rangle \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& [\![f]\!]_{\mathbb{N}} \circ \langle h_{B_1} \circ [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{M}}, \ldots, h_{B_n} \circ [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& [\![f]\!]_{\mathbb{N}} \circ h_{B_1} \times \cdots \times h_{B_n} \circ \langle [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{M}}, \ldots, [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ (MIso-Components) } \} \\
& [\![f]\!]_{\mathbb{N}} \circ h_{B_1 \times \cdots \times B_n} \circ \langle [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{M}}, \ldots, [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ (MHom-Func) } \} \\
& h_A \circ [\![f]\!]_{\mathbb{M}} \circ \langle [\![\Gamma \vdash u_1 : B_1]\!]_{\mathbb{M}}, \ldots, [\![\Gamma \vdash u_n : B_n]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ func rule } \} \\
& h_A \circ [\![\Gamma \vdash f(u_1, \ldots, u_n) : A]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

abs case Then $t = \lambda x{:}B.u$ and $A = B \to C$ for some $\Gamma + [x{:}B] \vdash u : C$, and

$$
\begin{aligned}
& [\![\Gamma \vdash \lambda x{:}B.u : B \to C]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ abs rule } \} \\
& \mathsf{curry}\left([\![\Gamma + [x{:}B] \vdash u : C]\!]_{\mathbb{N}} \right) \circ h_\Gamma \\
= \quad & \{ \text{ naturality of exponential } \} \\
& \mathsf{curry}\left([\![\Gamma + [x{:}B] \vdash u : C]\!]_{\mathbb{N}} \circ h_\Gamma \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}} \right) \\
= \quad & \{ \text{ inverses cancel } \} \\
& \mathsf{curry}\left([\![\Gamma + [x{:}B] \vdash u : C]\!]_{\mathbb{N}} \circ h_\Gamma \times (h_B \circ h_B^{-1}) \right) \\
= \quad & \{ \text{ } - \times - \text{ bifunctorial } \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{N}} \circ h_\Gamma \times h_B \circ \mathrm{id}_{\llbracket \Gamma \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right) \\
= \quad & \{\ \text{(MIso-Components)}\ \} \\
& \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{N}} \circ h_{\Gamma \times B} \circ \mathrm{id}_{\llbracket \Gamma \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right) \\
= \quad & \{\ \text{inductive hypothesis}\ \} \\
& \mathsf{curry}\left(h_C \circ \llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}} \circ \mathrm{id}_{\llbracket \Gamma \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right) \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& \mathsf{curry}\left(h_C \circ \mathsf{uncurry}\left(\mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right)\right) \circ \mathrm{id}_{\llbracket \Gamma \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right) \\
= \quad & \{\ \mathsf{uncurry}\ \} \\
& \mathsf{curry}\left(h_C \circ \mathsf{ev}_{\llbracket B \rrbracket_{\mathbb{M}}, \llbracket C \rrbracket_{\mathbb{M}}} \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right) \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}}} \circ \mathrm{id}_{\llbracket \Gamma \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right) \\
= \quad & \{\ - \times - \text{ interchange}\ \} \\
& \mathsf{curry}\left(h_C \circ \mathsf{ev}_{\llbracket B \rrbracket_{\mathbb{M}}, \llbracket C \rrbracket_{\mathbb{M}}} \circ \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}} \Rightarrow \llbracket C \rrbracket_{\mathbb{M}}} \times h_B^{-1} \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right) \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{N}}}\right) \\
= \quad & \{\ h_B^{-1} \Rightarrow h_C = \mathsf{curry}\left(h_C \circ \mathsf{ev}_{\llbracket B \rrbracket_{\mathbb{M}}, \llbracket C \rrbracket_{\mathbb{M}}} \circ \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{M}} \Rightarrow \llbracket C \rrbracket_{\mathbb{M}}} \times h_B^{-1}\right), \text{universal property of exponential}\ \} \\
& \mathsf{curry}\left(\mathsf{uncurry}\left(h_B^{-1} \Rightarrow h_C\right) \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right) \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{N}}}\right) \\
= \quad & \{\ \mathsf{uncurry}\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{\llbracket B \rrbracket_{\mathbb{N}}, \llbracket C \rrbracket_{\mathbb{N}}} \circ h_B^{-1} \Rightarrow h_C \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{N}}} \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right) \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{N}}}\right) \\
= \quad & \{\ - \times \llbracket B \rrbracket_{\mathbb{N}} \text{ functorial}\ \} \\
& \mathsf{curry}\left(\mathsf{ev}_{\llbracket B \rrbracket_{\mathbb{N}}, \llbracket C \rrbracket_{\mathbb{N}}} \circ \left(h_B^{-1} \Rightarrow h_C \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right)\right) \times \mathrm{id}_{\llbracket B \rrbracket_{\mathbb{N}}}\right) \\
= \quad & \{\ \mathsf{uncurry}\ \} \\
& \mathsf{curry}\left(\mathsf{uncurry}\left(h_B^{-1} \Rightarrow h_C \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right)\right)\right) \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& h_B^{-1} \Rightarrow h_C \circ \mathsf{curry}\left(\llbracket \Gamma + [x{:}B] \vdash u : C \rrbracket_{\mathbb{M}}\right) \\
= \quad & \{\ \text{(MIso-Components), abs rule}\ \} \\
& h_{B \to C} \circ \llbracket \Gamma \vdash \lambda x{:}B.u : B \to C \rrbracket_{\mathbb{M}}.
\end{aligned}
$$

◁

app case Then $t = uv$ for some $\Gamma \vdash u : B \to A$ and $\Gamma \vdash v : B$, so

$$
\begin{aligned}
& \llbracket \Gamma \vdash uv : A \rrbracket_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{\ \text{app rule}\ \}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{N}'}, [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \circ h_\Gamma \\
= \quad & \{ \text{ composition distributes over product } \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{N}} \circ h_\Gamma, [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \circ h_\Gamma \rangle \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle h_{B\to A} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}'}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ (MIso-Components) } \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle h_B^{-1} \Rightarrow h_A \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}'}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \ h_B^{-1} \Rightarrow h_A = \mathsf{curry}\left(h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1}\right) \ \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle \mathsf{curry}\left(h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1}\right) \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}'}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ naturality of exponential } \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \langle \mathsf{curry}\left(h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}}\right), h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& \mathsf{ev}_{[\![B]\!]_{\mathbb{N}'}[\![A]\!]_{\mathbb{N}}} \circ \mathsf{curry}\left(h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}}\right) \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}} \\
& \quad \circ \langle \mathrm{id}_{[\![\Gamma]\!]_{\mathbb{M}}}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ uncurry } \} \\
& \mathsf{uncurry}\left(\mathsf{curry}\left(h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}}\right)\right) \\
& \quad \circ \langle \mathrm{id}_{[\![\Gamma]\!]_{\mathbb{M}}}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ universal property of exponential } \} \\
& h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \mathrm{id}_{[\![B]\!]_{\mathbb{M}} \Rightarrow [\![A]\!]_{\mathbb{M}}} \times h_B^{-1} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \times \mathrm{id}_{[\![B]\!]_{\mathbb{N}}} \circ \langle \mathrm{id}_{[\![\Gamma]\!]_{\mathbb{M}}}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \ - \times - \text{ bifunctorial } \} \\
& h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \times h_B^{-1} \circ \langle \mathrm{id}_{[\![\Gamma]\!]_{\mathbb{M}}}, h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \langle [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}} \circ \mathrm{id}_{[\![\Gamma]\!]_{\mathbb{M}}}, h_B^{-1} \circ h_B \circ [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ identity cancels through composition, inverses cancel } \} \\
& h_A \circ \mathsf{ev}_{[\![B]\!]_{\mathbb{M}'}[\![A]\!]_{\mathbb{M}}} \circ \langle [\![\Gamma \vdash u : B \to A]\!]_{\mathbb{M}'}, [\![\Gamma \vdash v : B]\!]_{\mathbb{N}} \rangle \\
= \quad & \{ \text{ app rule } \} \\
& h_A \circ [\![\Gamma \vdash uv : A]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

pair case Then $t = \langle u, v \rangle$ and $A = B \times C$ for some $\Gamma \vdash u : B$ and $\Gamma \vdash v : C$, and

$$
\begin{aligned}
& [\![\Gamma \vdash \langle u, v \rangle : B \times C]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ pair rule } \}
\end{aligned}
$$

$$
\begin{aligned}
& \langle [\![\Gamma \vdash u : B]\!]_{\mathbb{N}}, [\![\Gamma \vdash v : C]\!]_{\mathbb{N}} \rangle \circ h_\Gamma \\
= \quad & \{ \text{ composition distributes over product } \} \\
& \langle [\![\Gamma \vdash u : B]\!]_{\mathbb{N}} \circ h_\Gamma, [\![\Gamma \vdash v : C]\!]_{\mathbb{N}} \circ h_\Gamma \rangle \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& \langle h_B \circ [\![\Gamma \vdash u : B]\!]_{\mathbb{M}}, h_C \circ [\![\Gamma \vdash v : C]\!]_{\mathbb{M}} \rangle \\
= \quad & \{ \text{ composition distributes over product } \} \\
& h_B \times h_C \circ \langle [\![\Gamma \vdash u : B]\!]_{\mathbb{M}}, [\![\Gamma \vdash v : C]\!]_{\mathbb{M}} \rangle \\
= \quad & \{ \text{ (MIso-Components) } \} \\
& h_{B \times C} \circ \langle [\![\Gamma \vdash u : B]\!]_{\mathbb{M}}, [\![\Gamma \vdash v : C]\!]_{\mathbb{M}} \rangle \\
= \quad & \{ \text{ pair rule } \} \\
& h_{B \times C} \circ [\![\langle u, v \rangle : B \times C]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

fst, snd case Then $t = \mathsf{fst}\,(u)$ for some B such that $\Gamma \vdash u : A \times B$, and

$$
\begin{aligned}
& [\![\Gamma \vdash \mathsf{fst}\,(u) : A]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ fst rule } \} \\
& \pi_1 \circ [\![\Gamma \vdash u : A \times B]\!]_{\mathbb{N}} \circ h_\Gamma \\
= \quad & \{ \text{ inductive hypothesis } \} \\
& \pi_1 \circ h_{A \times B} \circ [\![\Gamma \vdash u : A \times B]\!]_{\mathbb{M}} \\
= \quad & \{ \text{ (MIso-Components) } \} \\
& \pi_1 \circ h_A \times h_B \circ [\![\Gamma \vdash u : A \times B]\!]_{\mathbb{M}} \\
= \quad & \{ \text{ action of } - \times - \text{ bifunctor on morphisms } \} \\
& \pi_1 \circ \langle h_A \circ \pi_1, h_B \circ \pi_2 \rangle \circ [\![\Gamma \vdash u : A \times B]\!]_{\mathbb{M}} \\
= \quad & \{ \text{ universal property of product } \} \\
& h_A \circ \pi_1 \circ [\![\Gamma \vdash u : A \times B]\!]_{\mathbb{M}} \\
= \quad & \{ \text{ fst rule } \} \\
& h_A \circ [\![\Gamma \vdash \mathsf{fst}\,(u) : A]\!]_{\mathbb{M}}.
\end{aligned}
$$

◁

This establishes that $\mathbf{Ap}_{\mathbb{G}}^{-1}(h)$ is a natural transformation; given that h is a model isomorphism, each component of $\mathbf{Ap}_{\mathbb{G}}^{-1}(h)$ is an isomorphism, and so it is a natural isomorphism.

□

Proposition 4.32. *Every Cartesian closed category is equivalent to the syntactic category of its internal language:*

$$\mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right) \cong \mathscr{C}.$$

Proof. Let $\mathbb{M}$ be the canonical model of $\mathbf{Lan}\left(\mathscr{C}\right)$ in $\mathscr{C}$. Define a functor $F\colon\ \mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right) \to \mathscr{C}$, which is given on objects by $F(A) \coloneqq [\![A]\!] = A$, and on morphisms by

$$F([\![\ulcorner t \urcorner^A]\!]_{\mathfrak{I}}) \coloneqq \mathsf{uncurry}\left(\ldots \mathsf{uncurry}\left([\![\vdash t : A]\!]\right)\right) \quad \text{(the maximal uncurrying).}$$

The action of F is to effectively 'undo' all of the abstractions to yield a morphism with the domain equal to the interpretation of the product of the inputs to said abstractions.

Now define $G\colon \mathscr{C} \to \mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right)$, mapping $X \mapsto X$ for all objects X of $\mathscr{C}$, and

$$(f\colon X_1 \times \cdots \times X_n \to Y) \mapsto [\ulcorner \lambda x_1{:}X_1.\ \ldots\ .\lambda x_n{:}X_n.f(x_1, \ldots, x_n)^{\urcorner X_1 \to \cdots \to X_n \to Y}]_{\mathcal{T}}$$

for morphisms.

We will show that F and G when composed are naturally isomorphic to identity functors, i.e. finding natural isomorphisms

$$\epsilon\colon F \circ G \tilde{\Rightarrow} \mathrm{id}_{\mathscr{C}} \quad \text{and} \quad \eta\colon G \circ F \tilde{\Rightarrow} \mathrm{id}_{\mathbf{Syn}\left(\mathbf{Lan}\left(\mathscr{C}\right)\right)},$$

which serve witness to the equivalence.

For ϵ, on its components given a morphism $f\colon X_1 \times \cdots \times X_n \to Y$, we need to satisfy

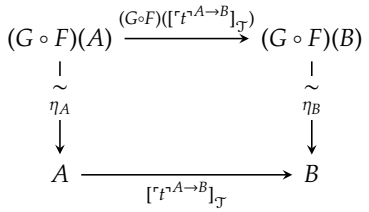
$$\begin{array}{ccc}
(F \circ G)(X_1 \times \cdots \times X_n) & \xrightarrow{(F\circ G)(f)} & (F \circ G)(Y) \\
\Big| \wr & & \Big| \wr \\
{\scriptstyle \epsilon_{X_1 \times \cdots \times X_n}}\downarrow & & {\scriptstyle \epsilon_Y}\downarrow \\
X_1 \times \cdots \times X_n & \xrightarrow[f]{} & Y
\end{array}$$

Observe that $(F \circ G)(X) = F(X) = [\![X]\!] = X$ for any object X of $\mathscr{C}$, and

$$\begin{aligned}
& (F \circ G)(f\colon X_1 \times \cdots \times X_n \to Y) \\
= \quad & \{\ \text{action of } G \text{ on morphisms}\ \} \\
& F([\ulcorner \lambda x_1{:}X_1.\ \ldots\ .\lambda x_n{:}X_n.f(x_1, \ldots, x_n)^{\urcorner X_1 \to \cdots \to X_n \to Y}]_{\mathcal{T}}) \\
= \quad & \{\ \text{action of } F \text{ on morphisms}\ \} \\
& \mathsf{uncurry}\left(\ldots \mathsf{uncurry}\left([\![\vdash \lambda x_1{:}X_1.\ \ldots\ .\lambda x_n{:}X_n.f(x_1, \ldots, x_n : X_1 \to \cdots \to X_n \to Y)]\!]\right)\right) \\
= \quad & \{\ \text{abs rule applied until codomain is } Y\ \} \\
& \mathsf{uncurry}\left(\ldots \mathsf{uncurry}\left(\mathsf{curry}\left(\ldots \mathsf{curry}\left([\![[x_1{:}X_1, \ldots, x_n{:}X_n] \vdash f(x_1, \ldots, x_n : Y)]\!]\right)\right)\right)\right) \\
= \quad & \{\ \text{universal property of exponential}\ \} \\
& [\![[x_1{:}X_1, \ldots, x_n{:}X_n] \vdash f(x_1, \ldots, x_n : Y)]\!] \\
= \quad & \{\ \text{func rule}\ \} \\
& [\![f]\!] \circ \langle [\![[x_1{:}X_1, \ldots, x_n{:}X_n] \vdash x_1 : X_1]\!], \ldots, [\![[x_1{:}X_1, \ldots, x_n{:}X_n] \vdash x_n : X_n]\!] \rangle \\
= \quad & \{\ \text{var rule}\ \} \\
& [\![f]\!] \circ \langle \pi_1, \ldots, \pi_n \rangle \\
= \quad & \{\ \text{canonicity of model, universal property of product}\ \} \\
& f.
\end{aligned}$$

So if we use identity morphisms for components, then the whole diagram degenerates and the condition is trivially satisfied.

For the reverse direction, η, on its components given an equivalence class of closed terms $[\ulcorner t^{\urcorner A \to B}]_{\mathcal{T}}$, we require



$(G \circ F)(A) = G(\llbracket A \rrbracket) = \llbracket A \rrbracket$ and similarly for B. By η-equivalence, we have $\vdash t = \lambda x{:}A.tx : A \to B$, so by the Soundness Theorem we have $\llbracket \vdash t : A \to B \rrbracket = \llbracket \vdash \lambda x{:}A.tx : A \to B \rrbracket$; using the abs rule, we discern that $\mathsf{uncurry}\left(\llbracket \vdash t : A \to B \rrbracket\right) = \llbracket [x{:}A] \vdash tx : B \rrbracket$, hence letting $s = \llbracket [x{:}A] \vdash tx : B \rrbracket \colon \llbracket A \rrbracket \to \llbracket B \rrbracket$,

$$(G \circ F)([\ulcorner t \urcorner^{A \to B}]_{\mathcal{T}}) = G(s) = [\ulcorner \lambda x{:}\llbracket A \rrbracket.s(x) \urcorner^{\llbracket A \rrbracket \to \llbracket B \rrbracket}]_{\mathcal{T}}.$$

We want find η subject to the commutativity of the diagram, which is that

$$\eta_{\llbracket B \rrbracket} \circ [\ulcorner \lambda x{:}\llbracket A \rrbracket.s(x) \urcorner^{\llbracket A \rrbracket \to \llbracket B \rrbracket}]_{\mathcal{T}} = [\ulcorner t \urcorner^{A \to B}]_{\mathcal{T}} \circ \eta_{\llbracket A \rrbracket}.$$

Define $\eta_{\llbracket X \rrbracket} \coloneqq [\ulcorner \mathsf{unmodel}_X \urcorner^{\llbracket X \rrbracket \to X}]_{\mathcal{T}}$; then this equation is equivalent to

$$[\ulcorner \lambda x{:}\llbracket A \rrbracket.\mathsf{unmodel}_B(s(x)) \urcorner^{\llbracket A \rrbracket \to B}]_{\mathcal{T}} = [\ulcorner \lambda x{:}\llbracket A \rrbracket.t(\mathsf{unmodel}_A(x)) \urcorner^{\llbracket A \rrbracket \to B}]_{\mathcal{T}}.$$

We can prove this by observing that in the canonical model,

$$\begin{aligned}
& \llbracket \vdash \lambda x{:}\llbracket A \rrbracket.\mathsf{unmodel}_B(s(x)) : \llbracket A \rrbracket \to B \rrbracket \\
= \quad & \{ \text{ abs rule } \} \\
& \mathsf{curry}\left(\llbracket [x{:}\llbracket A \rrbracket] \vdash \mathsf{unmodel}_B(s(x)) : B \rrbracket\right) \\
= \quad & \{ \text{ func rule } \} \\
& \mathsf{curry}\left(\llbracket \mathsf{unmodel}_B \rrbracket \circ \llbracket [x{:}\llbracket A \rrbracket] \vdash s(x) : \llbracket B \rrbracket \rrbracket\right) \\
= \quad & \{ \text{ canonicity of model } \} \\
& \mathsf{curry}\left(\mathrm{id}_B \circ \llbracket [x{:}\llbracket A \rrbracket] \vdash s(x) : \llbracket B \rrbracket \rrbracket\right) \\
= \quad & \{ \text{ identities cancel through composition } \} \\
& \mathsf{curry}\left(\llbracket [x{:}\llbracket A \rrbracket] \vdash s(x) : \llbracket B \rrbracket \rrbracket\right) \\
= \quad & \{ \text{ func rule } \} \\
& \mathsf{curry}\left(\llbracket s \rrbracket \circ \llbracket [x{:}\llbracket A \rrbracket] \vdash x : \llbracket A \rrbracket \rrbracket\right) \\
= \quad & \{ \text{ var rule, universal property of product } \} \\
& \mathsf{curry}\left(\llbracket s \rrbracket \circ \mathrm{id}_{\llbracket A \rrbracket}\right) \\
= \quad & \{ \text{ identities cancel through composition } \} \\
& \mathsf{curry}\left(\llbracket s \rrbracket\right) \\
= \quad & \{ \text{ canonicity of model } \} \\
& \mathsf{curry}\,(s) \\
= \quad & \{ \; s = \llbracket [x{:}A] \vdash t(x) : B \rrbracket \; \}
\end{aligned}$$

$$
\begin{aligned}
& \mathsf{curry}\left(\llbracket [x{:}A] \vdash t(x) : B \rrbracket\right) \\
= \quad & \{ \text{ func rule } \} \\
& \mathsf{curry}\left(\llbracket t \rrbracket \circ \llbracket [x{:}A] \vdash x : A \rrbracket\right) \\
= \quad & \{ \text{ identities cancel through composition } \} \\
& \mathsf{curry}\left(\llbracket t \rrbracket \circ \mathrm{id}_A \circ \llbracket [x{:}A] \vdash x : A \rrbracket\right) \\
= \quad & \{ \text{ canonicity of model } \} \\
& \mathsf{curry}\left(\llbracket t \rrbracket \circ \llbracket \mathsf{unmodel}_A \rrbracket \circ \llbracket [x{:}\llbracket A \rrbracket] \vdash x : \llbracket A \rrbracket \rrbracket\right) \\
= \quad & \{ \text{ func rule } \} \\
& \mathsf{curry}\left(\llbracket t \rrbracket \circ \llbracket [x{:}\llbracket A \rrbracket] \vdash \mathsf{unmodel}_A(x) : A \rrbracket\right) \\
= \quad & \{ \text{ func rule } \} \\
& \mathsf{curry}\left(\llbracket [x{:}\llbracket A \rrbracket] \vdash t(\mathsf{unmodel}_A(x)) : B \rrbracket\right) \\
= \quad & \{ \text{ abs rule } \} \\
& \llbracket \vdash \lambda x{:}\llbracket A \rrbracket . t(\mathsf{unmodel}_A(x)) : B \rrbracket,
\end{aligned}
$$

and this certainly implies that each component of η is an isomorphism.

□

REFERENCES

Abramsky, S., & Tzevelekos, N. (2010). Introduction to categories and categorical logic. In *New structures for physics*, Springer, pp. 3–94.
Awodey, S. (2010). *Category theory*, Oxford University Press.
Awodey, S., & Bauer, A. (2017). Lecture notes: Introduction to categorical logic.
Baader, F., & Nipkow, T. (1998). *Term rewriting and all that*, New York, NY, USA: Cambridge University Press.
Barendregt, H. (1984). *The lambda calculus: Its syntax and semantics*, North-Holland.
Ben-Yelles, C.-B. (1979). *Type assignment in the lambda-calculus: Syntax and semantics* (PhD thesis), University College of Swansea.
Böhm, C., & Gross, W. (1966). Introduction to the cuch. *Automata Theory*, 35–65.
Bruijn, N. G. de. (1972). Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes mathematicae (proceedings)*, Vol. 75, Elsevier, pp. 381–392.
Cardone, F., & Hindley, J. R. (2006). History of lambda-calculus and combinatory logic.
Church, A. (1932). A set of postulates for the foundation of logic. *Annals of Mathematics*, 346–366.
Church, A. (1936). A note on the entscheidungsproblem. *The Journal of Symbolic Logic*, **1**(1), 40–41.
Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic*, **5**(2), 56–68.
Crole, R. L. (1993). *Categories for types*, Cambridge University Press.
Felleisen, M., & Friedman, D. P. (1986). *Control operators, the secd-machine, and the [1]-calculus*, Indiana University, Computer Science Department.
Floyd, R. W. (1967). Assigning meanings to programs. *Mathematical Aspects of Computer Science*, **19**(19-32), 1.
Girard, J.-Y., Lafont, Y., & Taylor, P. (1989). *Proofs and types*, Vol. 7, Cambridge University Press Cambridge.
Harper, R. (2011). The holy trinity.
Harper, R. (2013). Advanced topics in programming languages. Retrieved from https://scs.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0945cc7f-48b7-4803-81af-e7193a3f461d
Harper, R. (2016). *Practical foundations for programming languages*, Cambridge University Press.
Hennessy, M. (1990). *The semantics of programming languages: An elementary introduction using structural operational semantics*, John Wiley & Sons.
Hindley, J. R. (1997). *Basic simple type theory*, Vol. 42, Cambridge University Press.
Hoare, C. A. R. (1969). An axiomatic basis for computer programming. *Communications of the ACM*, **12**(10), 576–580.
Jacobs, B. (1999). *Categorical logic and type theory*, Vol. 141, Elsevier.
Kleene, S. C., & Rosser, J. B. (1935). The inconsistency of certain formal logics. *Annals of Mathematics*, **36**(3), 630–636.
Leinster, T. (2016). Basic Category Theory. *ArXiv E-Prints*. Retrieved from http://arxiv.org/abs/1612.09375
Mac Lane, S. (1998). *Categories for the working mathematician*, Springer.
McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine, part i. *Communications of the ACM*, **3**(4), 184–195.
Milewski, B. (2017). Category theory for programmers. Retrieved 1 October 2017, from https://bartoszmilewski.com/2014/10/28/category-theory-for-programmers-the-preface/
O'donnell, M. J. (1977). Computing in systems described by equations.
Pierce, B. C. (1991). *Basic category theory for computer scientists*, MIT press.

Pierce, B. C. (2002). *Types and programming languages*, MIT press.
Pitts, A. M. (2001). Categorical logic, handbook of logic in computer science: Volume 5: Logic and algebraic methods, Oxford University Press, Oxford.
Plotkin, G. D. (1977). LCF considered as a programming language. *Theoretical Computer Science*, **5**(3), 223–255.
Reynolds, J. C. (1972). Definitional interpreters for higher-order programming languages. In *Proceedings of the acm annual conference-volume 2*, ACM, pp. 717–740.
Scott, D. (1970). *Outline of a mathematical theory of computation*, Oxford University Computing Laboratory, Programming Research Group.
Sipser, M. (2012). *Introduction to the theory of computation*, Cengage Learning.
Slonneger, K., & Kurtz, B. L. (1995). *Formal syntax and semantics of programming languages*, Vol. 340, Addison-Wesley Reading.
Stoy, J. E. (1977). *Denotational semantics: The scott-strachey approach to programming language theory*, Vol. 45, MIT press Cambridge.
Turing, A. M. (1937a). Computability and λ-definability. *The Journal of Symbolic Logic*, **2**(4), 153–163.
Turing, A. M. (1937b). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, **2**(1), 230–265.

INDEX