Toward the Automation of Category Theory

Dexter Kozen
Department of Computer Science
Cornell University
Ithaca, New York 14853-7501, USA
kozen@cs.cornell.edu

September 8, 2004

Abstract

We introduce a sequent system for basic category-theoretic reasoning suitable for computer implementation. We illustrate its use by giving a complete formal proof that the functor categories $\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]$ and $\mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]]$ are naturally isomorphic.

1 Introduction

Since its invention in 1945 by Samuel Eilenberg and Saunders Mac Lane [7], category theory has had a profound impact in many areas of mathematics and computer science; see for example [8, 9, 11, 13, 16].

Unfortunately, many basic category-theoretic facts, although easy to state, can be quite tedious to verify formally. For example, consider the well-known fact that Cat, the category of (small) categories, is cartesian-closed [10, Ex. 1, p. 45], or more generally, that there exists a natural isomorphism of functor categories

$$\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \quad \cong \quad \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]] \tag{1}$$

[10, Ex. 2, p. 45]. This statement is deceptively concise in that it contains a large amount of compressed information. A complete formal verification by hand would be quite onerous due to the enormous number of low-level details that must be checked. This task is uninteresting, and one would like to automate as much of it as possible.

In print, authors often do not bother to provide formal details of the constructions that establish such basic facts, let alone proofs of correctness. Those arguments that are explicitly given are typically expressed in terms of commuting diagrams, and verification amounts to visual arrow chasing. This is adequate for humans, but does not lend itself well to automation. Eklund et al. [8] present a graphical technique aimed at simplifying the verification of such category-theoretic constructions, but again this is meant for human consumption and not for computers.

There have been a few attempts at automating parts of category theory [3, 5, 14, 15]. The closest in spirit to the present work is the system of Cáccamo and Winskel [5], which we denote here by CW. Their system is a second-order sequent calculus in which types denote categories and expressions denote functors. Equational judgements are interpreted as natural isomorphisms between functors. The system can prove that a certain expressions are functorial in their free variables or that two expressions are naturally isomorphic. In addition, it is able to handle arguments involving limits and more general ends, thereby supporting algebraic manipulation of universal constructions. They suggest an approach to implementation of the calculus in the theorem prover Isabelle/HOL [12].

One omission in the work of Cáccamo and Winskel is machinery for reasoning about the more primitive building blocks on which the theory is based. In this paper we attempt to fill this gap. We present a first-order sequent calculus in the style of CW that captures the basic properties of categories, functors, and natural transformations. As with CW, our system is a mix of typing judgements and equational reasoning, but unlike CW, types are sets of objects and homsets of (small) categories, expressions represent objects and arrows, and the equations are interpreted as equality of objects and arrows, not isomorphisms. We have used the system to prove the natural isomorphism (1) mentioned above, among other basic category-theoretic facts. The system is thus complementary to CW and might coexist with it in a supporting role.

The chief differences between the two systems are:

(i) Our system is strictly first-order, whereas CW is primarily second-order. It is noteworthy that many basic category-theoretic facts can be established without resorting to any second-order constructs. This is important from both a complexity-theoretic and proof-theoretic standpoint.

(ii) CW contains no pure equality construct. Equations are interpreted as isomorphisms. In contrast, our system can reason about equality of expressions representing objects, arrows, functors, and natural transformations.

(iii) CW assumes several high-level theorems such as the Yoneda embedding as axioms in the form of proof rules, whereas our system operates at a more primitive level. It captures the fundamental definitions of functors and natural transformations in a few symmetric first-order introduction and elimination rules. It is quite concise compared to CW, and it is interesting to see just how far one can go with so little machinery.

(iv) The implementation as envisioned by Cáccamo and Winskel would be in the form of a proof assistant, where an expert user would direct the evolution of the proof. In contrast, after working out several examples in our system, it is apparent that a large part of the reasoning, if not all, can be fully automated. One observes that the application of rules is largely syntax-directed and deterministic. Modulo equational reasoning, arguments tend to break down into the application of analysis (elimination) rules followed by the application of synthesis (introduction) rules, suggesting a normal form for proofs. Equational reasoning involves primarily substitution of equals for equals and extensionality rules. The extensionality rules are not the general extensionality (η-) rules of the typed lambda calculus, but rather rules of a more limited first-order form (rules (18)-(21) below), and their use again tends to exhibit a discernable normal form.

As with CW, our system is quite amenable to computer implementation. This might be done in an existing framework such as Isabelle/HOL [12] or NuPrl [6], however the higher-order facilities of these systems would not be required.

2 Notational Conventions

We assume familiarity with the basic definitions and notation of category theory [2, 10]. To simplify notation, we will adhere to the following conventions.

- Symbols in sans-serif, such as C, always denote categories. The categories Set and Cat are the categories of sets and set functions and of (small) categories and functors, respectively.
- If C is a category, we use the symbol C to denote both the category C and the set of objects of C.
- We write $A : \mathsf{C}$ to indicate that A is an object of C. Composition is denoted by the symbol $\circ$ and the identity on object $A : \mathsf{C}$ is denoted $\mathsf{1}_A$. The use of a symbol in sans serif, such as C, implicitly carries the type assertion $\mathsf{C} : \mathsf{Cat}$.
- We write $h : \mathsf{C}(A, B)$ to indicate that h is an arrow of the category C with domain A and codomain B.
- $\mathsf{Fun}\,[\mathsf{C}, \mathsf{D}]$ denotes the functor category whose objects are functors from C to D and whose arrows are natural transformations on such functors. This is the same as the category denoted D^{C} in [10]. Thus $F : \mathsf{Fun}\,[\mathsf{C}, \mathsf{D}]$ indicates that F is a functor from C to D and $\varphi : \mathsf{Fun}\,[\mathsf{C}, \mathsf{D}]\,(F, G)$ indicates that φ is a natural transformation with domain F and codomain G, where $F, G : \mathsf{Fun}\,[\mathsf{C}, \mathsf{D}]$.
- C^{op} denotes the opposite category of C.
- $f : X \to Y$ indicates that $f : \mathsf{Set}(X, Y)$, that is, f is a set function from set X to set Y. We use the symbol $\to$ only in this context. Function application is written as juxtaposition and associates to the left.
- F^1 and F^2 denote the object and arrow components, respectively, of a functor F. Thus if $F : \mathsf{Fun}\,[\mathsf{C}, \mathsf{D}]$, $A, B : \mathsf{C}$, and $h : \mathsf{C}(A, B)$, then $F^1 A, F^1 B : \mathsf{D}$ and $F^2 h : \mathsf{D}(F^1 A, F^1 B)$.

- Function application binds tighter than the operators ^{1} and ^{2}. Thus the expression F^1A^2 should be parsed $(F^1A)^2$.
- $\mathsf{C} \times \mathsf{D}$ denotes the product of categories C and D. Its objects are pairs $(A, X) : \mathsf{C} \times \mathsf{D}$, where $A : \mathsf{C}$ and $X : \mathsf{D}$, and its arrows are pairs $(f, h) : (\mathsf{C} \times \mathsf{D})((A, X), (B, Y))$, where $f : \mathsf{C}(A, B)$ and $h : \mathsf{D}(X, Y)$. Composition and identities are defined componentwise; that is,

$$(g, k) \circ (f, h) \stackrel{\text{def}}{=} (g \circ f, k \circ h) \tag{2}$$

$$\mathsf{1}_{(A,X)} \stackrel{\text{def}}{=} (\mathsf{1}_A, \mathsf{1}_X). \tag{3}$$

- We sometimes omit the commas in triples, e.g.

$$(\mathsf{CDE}) \stackrel{\text{def}}{=} (\mathsf{C}, \mathsf{D}, \mathsf{E})$$

$$(PQR) \stackrel{\text{def}}{=} (P, Q, R).$$

3 Rules

The rules involve sequents $\Gamma \vdash \alpha$, where Γ is a type environment (set of type judgements on atomic symbols) and α is either a type judgement or an equation. There is a set of rules for functors and a set for natural transformations, as well as some rules covering the basic properties of categories and equational reasoning.

The rules for functors and natural transformations are the most interesting. They are divided into symmetric sets of rules for analysis (elimination) and synthesis (introduction).

Categories

There is a collection of rules covering the basic properties of categories, which are essentially the rules of typed monoids. These rules include typing rules for composition and identities

$$\frac{\Gamma \vdash A, B, C : \mathsf{C}, \quad \Gamma \vdash f : \mathsf{C}(A, B), \quad \Gamma \vdash g : \mathsf{C}(B, C)}{\Gamma \vdash g \circ f : \mathsf{C}(A, C)} \tag{4}$$

$$\frac{\Gamma \vdash A : \mathsf{C}}{\Gamma \vdash \mathsf{1}_A : \mathsf{C}(A, A)}, \tag{5}$$

as well as equational rules for associativity and two-sided identity.

Functors

A functor F from C to D is determined by its object and arrow components F^1 and F^2. The components must be of the correct type and must preserve composition and identities. These properties are captured in the following rules.

Analysis

$$\frac{\Gamma \vdash F : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma \vdash A : \mathsf{C}}{\Gamma \vdash F^1 A : \mathsf{D}} \tag{6}$$

$$\frac{\Gamma \vdash F : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma \vdash A, B : \mathsf{C}, \quad \Gamma \vdash f : \mathsf{C}(A,B)}{\Gamma \vdash F^2 f : \mathsf{D}(F^1 A, F^1 B)} \tag{7}$$

$$\frac{\Gamma \vdash F : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma \vdash A, B, C : \mathsf{C}, \quad \Gamma \vdash f : \mathsf{C}(A,B), \quad \Gamma \vdash g : \mathsf{C}(B,C)}{\Gamma \vdash F^2(g \circ f) = F^2 g \circ F^2 f} \tag{8}$$

$$\frac{\Gamma \vdash F : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma \vdash A : \mathsf{C}}{\Gamma \vdash F^2 \mathsf{1}_A = \mathsf{1}_{F^1 A}} \tag{9}$$

Synthesis

$$\frac{\begin{array}{l} \Gamma,\ A : \mathsf{C} \vdash F^1 A : \mathsf{D} \\ \Gamma,\ A, B : \mathsf{C},\ g : \mathsf{C}(A,B) \vdash F^2 g : \mathsf{D}(F^1 A, F^1 B) \\ \Gamma,\ A, B, C : \mathsf{C},\ f : \mathsf{C}(A,B),\ g : \mathsf{C}(B,C) \vdash F^2(g \circ f) = F^2 g \circ F^2 f \\ \Gamma,\ A : \mathsf{C} \vdash F^2 \mathsf{1}_A = \mathsf{1}_{F^1 A} \end{array}}{\Gamma \vdash F : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]} \tag{10}$$

Natural Transformations

A natural transformation $\varphi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F, G)$ is a function that for each object $A : \mathsf{C}$ gives an arrow $\varphi A : \mathsf{D}(F^1 A, G^1 A)$, called the *component* of φ at A, such that for all arrows $g : \mathsf{C}(A, B)$, the following diagram commutes:

$$\begin{tikzcd} F^1 A \arrow[r, "F^2 g"] \arrow[d, "\varphi A"'] & F^1 B \arrow[d, "\varphi B"] \\ G^1 A \arrow[r, "G^2 g"] & G^1 B \end{tikzcd} \tag{11}$$

Composition and identities are defined by

$$(\varphi \circ \psi) A \stackrel{\text{def}}{=} \varphi A \circ \psi A \tag{12}$$

$$\mathsf{1}_F A \stackrel{\text{def}}{=} \mathsf{1}_{F^1 A}. \tag{13}$$

The property (11), along with the typing of φ, are captured in the following rules.

Analysis

$$\frac{\Gamma \vdash \varphi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F,G)}{\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]} \tag{14}$$

$$\frac{\Gamma \vdash \varphi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F,G), \quad \Gamma \vdash A : \mathsf{C}}{\Gamma \vdash \varphi A : \mathsf{D}(F^1 A, G^1 A)} \tag{15}$$

$$\frac{\Gamma \vdash \varphi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F,G), \quad \Gamma \vdash A, B : \mathsf{C}, \quad \Gamma \vdash g : \mathsf{C}(A,B)}{\Gamma \vdash \varphi B \circ F^2 g = G^2 g \circ \varphi A} \tag{16}$$

Synthesis

$$\frac{\begin{array}{l}\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}] \\ \Gamma,\ A : \mathsf{C} \vdash \varphi A : \mathsf{D}(F^1 A, G^1 A) \\ \Gamma,\ A, B : \mathsf{C},\ g : \mathsf{C}(A,B) \vdash \varphi B \circ F^2 g = G^2 g \circ \varphi A\end{array}}{\Gamma \vdash \varphi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F,G)} \tag{17}$$

Equational Reasoning

The chief tool for equational reasoning is substitution of equals for equals. We also provide extensionality rules for objects of functional type:

$$\frac{\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma, A : \mathsf{C} \vdash F^1 A = G^1 A}{\Gamma \vdash F^1 = G^1} \tag{18}$$

$$\frac{\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma,\ A, B : \mathsf{C},\ g : \mathsf{C}(A,B) \vdash F^2 g = G^2 g}{\Gamma \vdash F^2 = G^2} \tag{19}$$

$$\frac{\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,, \quad \Gamma \vdash F^1 = G^1, \quad \Gamma \vdash F^2 = G^2}{\Gamma \vdash F = G} \tag{20}$$

$$\frac{\begin{array}{l}\Gamma \vdash F, G : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}] \\ \Gamma \vdash \varphi, \psi : \mathsf{Fun}\,[\mathsf{C},\mathsf{D}]\,(F,G) \\ \Gamma,\ A : \mathsf{C} \vdash \varphi A = \psi A\end{array}}{\Gamma \vdash \varphi = \psi} \tag{21}$$

Certain equations on objects and arrows are assumed as axioms, including the associativity of composition and two-sided identity rules for arrows, the equations (2) and (3) for products, and the equations (12) and (13) for natural transformations.

We also allow equations on types and substitution of equals for equals in type expressions. Any such equation $\alpha = \beta$ takes the form of a rule

$$\frac{\Gamma \vdash A : \alpha}{\Gamma \vdash A : \beta}. \tag{22}$$

We postulate as axioms the type equations

$$\mathsf{Cat}(\mathsf{C},\mathsf{D}) = \mathsf{Fun}\,[\mathsf{C},\mathsf{D}] \tag{23}$$

$$\mathsf{C} = \mathsf{C}^{\mathsf{op}} \tag{24}$$

$$\mathsf{C}(A,B) = \mathsf{C}^{\mathsf{op}}(B,A). \tag{25}$$

Other Rules

There are also various rules for products, weakening, and other structural rules; see [5]. These are quite standard and do not bear explicit mention.

4 An Application

We illustrate the system by giving a formal proof of the following fact.

Theorem 4.1 ([10, Ex. 2, p. 45]) *Let* $\mathsf{C}, \mathsf{D}, \mathsf{E}$ *be categories. Then*

$$\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \quad \cong \quad \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]]\,,$$

where $\cong$ *denotes isomorphism of categories. The isomorphism is natural in* C, D*, and* E*.*

Proof. The proof will be broken into four steps:

1. the construction of a functor

$$\theta(\mathsf{CDE}) \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,, \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]]]\,;$$

2. the construction of a functor

$$\eta(\mathsf{CDE}) \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]]\,, \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]]\,;$$

3. the demonstration that $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$ are inverses;

4. establishing naturality.

In step 4, we show that the functors $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$ are natural in the parameters $\mathsf{C}, \mathsf{D}, \mathsf{E}$; that is, θ and η are natural transformations with components $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$. However, we will not need to make the dependence on (CDE) explicit until step 4, so to save notation, we will write θ for $\theta(\mathsf{CDE})$ and η for $\eta(\mathsf{CDE})$ in steps 1–3.

Step 1 For this step, we will work in the following type environment:

$$\begin{array}{rcl}
F, G, H & : & \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \\
A, B, C & : & \mathsf{C} \\
X, Y, Z & : & \mathsf{D} \\
f & : & \mathsf{C}(A, B) \\
g & : & \mathsf{C}(B, C) \\
h & : & \mathsf{D}(X, Y) \\
k & : & \mathsf{D}(Y, Z) \\
\varphi & : & \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(F, G) \\
\psi & : & \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(G, H).
\end{array}$$

Under these assumptions, define

$$\begin{aligned}
\theta^1 F^1 A^1 X &\stackrel{\text{def}}{=} F^1(A,X) : \mathsf{E} &(26)\\
\theta^1 F^1 A^2 h &\stackrel{\text{def}}{=} F^2(1_A,h) : \mathsf{E}(F^1(A,X),F^1(A,Y)) &(27)\\
\theta^1 F^2 f X &\stackrel{\text{def}}{=} F^2(f,1_X) : \mathsf{E}(F^1(A,X),F^1(B,X)) &(28)\\
\theta^2 \varphi A X &\stackrel{\text{def}}{=} \varphi(A,X) : \mathsf{E}(F^1(A,X),G^1(A,X)). &(29)
\end{aligned}$$

The type judgement in (26) follows from (6); in (27) and (28), from (5) and (7); and in (29), from (15). We must verify that the definitions (26)–(29) define a functor θ of the correct type.

It follows from (26)–(29) that

$$\begin{aligned}
\theta^1 F^1 A^2 h &: \mathsf{E}(\theta^1 F^1 A^1 X, \theta^1 F^1 A^1 Y) &(30)\\
\theta^1 F^2 f X &: \mathsf{E}(\theta^1 F^1 A^1 X, \theta^1 F^1 B^1 X) &(31)\\
\theta^2 \varphi A X &: \mathsf{E}(\theta^1 F^1 A^1 X, \theta^1 G^1 A^1 X). &(32)
\end{aligned}$$

Also, using (27) and the basic equational properties of functors and composition,

$$\begin{aligned}
\theta^1 F^1 A^2 (g \circ h) &= F^2(1_A, g\circ h) \;=\; F^2(1_A,g)\circ F^2(1_A,h)\\
&= \theta^1 F^1 A^2 g \circ \theta^1 F^1 A^2 h &(33)\\
\theta^1 F^1 A^2 1_X &= F^2(1_A,1_X) \;=\; F^2(1_{(A,X)})\\
&= 1_{F^1(A,X)} \;=\; 1_{\theta^1 F^1 A^1 X}. &(34)
\end{aligned}$$

Applying the rule (10) with premises (26), (30), (33), and (34), we have

$$\theta^1 F^1 A \quad : \quad \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,. \tag{35}$$

Since $A : \mathsf{C}$ was arbitrary, the conclusion (35) essentially says that the object component $\theta^1 F^1$ of $\theta^1 F$ is a function of type $\mathsf{C} \to \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]$. We cannot express this, since the type constructor $\to$ is not part of the language. Nevertheless, it is enough to show what we have shown in order to establish a premise in the application below of the synthesis rules (10) and (17).

For the arrow component $\theta^1 F^2$ of $\theta^1 F$, using (27) and (28),

$$\begin{aligned}
\theta^1 F^2 f Y \circ \theta^1 F^1 A^2 h &= F^2(f,1_Y)\circ F^2(1_A,h) \;=\; F^2(f,h)\\
&= F^2(1_B,h)\circ F^2(f,1_X)\\
&= \theta^1 F^1 B^2 h \circ \theta^1 F^2 f X. &(36)
\end{aligned}$$

Applying the rule (17) with premises (35), (31), and (36),

$$\theta^1 F^2 f \quad : \quad \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(\theta^1 F^1 A, \theta^1 F^1 B). \tag{37}$$

Again, since $f : \mathsf{C}(A,B)$ was arbitrary, we have essentially shown that $\theta^1 F^2$ is a function whose type in a higher-order system would be expressed

$$\forall A : \mathsf{C}\ .\ \forall B : \mathsf{C}\ .\ \mathsf{C}(A,B) \to \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(\theta^1 F^1 A, \theta^1 F^1 B),$$

but our first-order language cannot express this.

By (28), (8), and (12),

$$\begin{aligned}\theta^1 F^2(g \circ f)X &= F^2(g \circ f, 1_X) = F^2(g, 1_X) \circ F^2(f, 1_X) \\ &= \theta^1 F^2 gX \circ \theta^1 F^2 fX = (\theta^1 F^2 g \circ \theta^1 F^2 f)X,\end{aligned}$$

and by (28), (9), (26), and (13),

$$\begin{aligned}\theta^1 F^2 1_A X &= F^2(1_A, 1_X) = F^2 1_{(A,X)} \\ &= 1_{F^1(A,X)} = 1_{\theta^1 F^1 A^1 X} = 1_{\theta^1 F^1 A} X,\end{aligned}$$

therefore by extensionality (21),

$$\theta^1 F^2(g \circ f) = \theta^1 F^2 g \circ \theta^1 F^2 f, \tag{38}$$

$$\theta^1 F^2 1_A = 1_{\theta^1 F^1 A}. \tag{39}$$

Now applying the rule (10) with premises (35), (37), (38), and (39),

$$\theta^1 F \quad : \quad \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,. \tag{40}$$

Since $F : \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]$ was arbitrary, this will imply that the object component θ^1 of θ is of the correct type.

For the arrow component θ^2, by (29), (27), and (16),

$$\begin{aligned}\theta^2 \varphi A Y \circ \theta^1 F^1 A^2 h &= \varphi(A, Y) \circ F^2(1_A, h) \\ &= G^2(1_A, h) \circ \varphi(A, X) \\ &= \theta^1 G^1 A^2 h \circ \theta^2 \varphi A X.\end{aligned} \tag{41}$$

Using rule (17) with premises (35), (32), and (41),

$$\theta^2 \varphi A \quad : \quad \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,(\theta^1 F^1 A, \theta^1 G^1 A). \tag{42}$$

In addition, using (29), (28), and (16),

$$\begin{aligned}(\theta^2 \varphi B \circ \theta^1 F^2 f)X &= \theta^2 \varphi B X \circ \theta^1 F^2 f X = \varphi(B, X) \circ F^2(f, 1_X) \\ &= G^2(f, 1_X) \circ \varphi(A, X) = \theta^1 G^2 f X \circ \theta^2 \varphi A X \\ &= (\theta^1 G^2 f \circ \theta^2 \varphi A)X.\end{aligned} \tag{43}$$

Since $X : \mathsf{D}$ was arbitrary, by (21),

$$\theta^2 \varphi B \circ \theta^1 F^2 f = \theta^1 G^2 f \circ \theta^2 \varphi A. \tag{44}$$

It follows from rule (17) with (40), (42), and (44) as premises that

$$\theta^2 \varphi \quad : \quad \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,(\theta^1 F, \theta^1 G). \tag{45}$$

Using (29), (26), and the basic properties of natural transformations (12) and (13),

$$\begin{array}{rcl} \theta^2(\psi\circ\varphi)AX &=& (\psi\circ\varphi)(A,X) \;\;=\;\; \psi(A,X)\circ\varphi(A,X) \\ &=& \theta^2\psi AX\circ\theta^2\varphi AX \;\;=\;\; (\theta^2\psi\circ\theta^2\varphi)AX, \\ \theta^2 1_F AX &=& 1_F(A,X) \;\;=\;\; 1_{F^1(A,X)} \;\;=\;\; 1_{\theta^1 F^1 A^1 X} \\ &=& 1_{\theta^1 F^1 A}X \;\;=\;\; 1_{\theta^1 F}AX. \end{array}$$

Since $A : \mathsf{C}$ and $X : \mathsf{D}$ were arbitrary, by two applications of extensionality (21), we have

$$\theta^2(\psi\circ\varphi) \;=\; \theta^2\psi\circ\theta^2\varphi \tag{46}$$

$$\theta^2 1_F \;=\; 1_{\theta^1 F}. \tag{47}$$

It follows from rule (10) with (40), (45), (46), and (47) as premises that

$$\theta \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C}\times\mathsf{D},\mathsf{E}]\,,\mathsf{Fun}\,[\mathsf{C},\mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,]\,]\,.$$

This establishes that θ is a functor of the correct type.

Step 2 For this step, we will work in the following type environment:

$$\begin{array}{rcl} F,G,H &:& \mathsf{Fun}\,[\mathsf{C},\mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,] \\ A,B,C &:& \mathsf{C} \\ X,Y,Z &:& \mathsf{D} \\ f &:& \mathsf{C}(A,B) \\ g &:& \mathsf{C}(B,C) \\ h &:& \mathsf{D}(X,Y) \\ k &:& \mathsf{D}(Y,Z) \\ \varphi &:& \mathsf{Fun}\,[\mathsf{C},\mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,]\,(F,G) \\ \psi &:& \mathsf{Fun}\,[\mathsf{C},\mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,]\,(G,H). \end{array}$$

Under these assumptions, define

$$\eta^1 F^1(A,X) \;\stackrel{\text{def}}{=}\; F^1A^1X : \mathsf{E} \tag{48}$$

$$\eta^1 F^2(f,h) \;\stackrel{\text{def}}{=}\; F^2fY\circ F^1A^2h \;\;=\;\; F^1B^2h\circ F^2fX \tag{49}$$

$$\eta^2\varphi(A,X) \;\stackrel{\text{def}}{=}\; \varphi AX : \mathsf{E}(F^1A^1X, G^1A^1X) \tag{50}$$

(we will argue that the two terms on the right-hand side of (49) are equal and discuss their typing below). The type judgement in (48) follows from two applications of (6); and in (50), from two applications of (15). It follows from (48) and (50) that

$$\eta^2\varphi(A,X) \quad : \quad \mathsf{E}(\eta^1F^1(A,X),\eta^1G^1(A,X)). \tag{51}$$

First we show that the types of $\eta^1 F^1$ and $\eta^1 F^2$ are correct. For $\eta^1 F^1$, since $A : \mathsf{C}$ and $X : \mathsf{D}$ were arbitrary, (48) will imply that $\eta^1 F^1$ is of the correct type.

For $\eta^1 F^2$, by (6), (7), and (48), we have

$$\begin{array}{rcl} F^1 A^2 h & : & \mathsf{E}(\eta^1 F^1(A, X), \eta^1 F^1(A, Y)) \\ F^1 B^2 h & : & \mathsf{E}(\eta^1 F^1(B, X), \eta^1 F^1(B, Y)). \end{array}$$

Also, by (7), (15), and (48), we have

$$\begin{array}{rcl} F^2 f X & : & \mathsf{E}(\eta^1 F^1(A, X), \eta^1 F^1(B, X)) \\ F^2 f Y & : & \mathsf{E}(\eta^1 F^1(A, Y), \eta^1 F^1(B, Y)). \end{array}$$

Thus for $(f, h) : (\mathsf{C} \times \mathsf{D})((A, X), (B, Y))$,

$$F^2 f Y \circ F^1 A^2 h,\ F^1 B^2 h \circ F^2 f X : \mathsf{E}(\eta^1 F^1(A, X), \eta^1 F^1(B, Y)).$$

Since $F^2 f : \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,(F^1 A, F^1 B)$ by (7),

$$F^2 f Y \circ F^1 A^2 h \quad = \quad F^1 B^2 h \circ F^2 f X$$

by (16), therefore $\eta^1 F^2(f, h)$ is well defined by (49), and

$$\eta^1 F^2(f, h) \quad : \quad \mathsf{E}(\eta^1 F^1(A, X), \eta^1 F^1(B, Y)). \tag{52}$$

To show that $\eta^1 F^2$ respects composition and identities, we reason equationally.

$$\begin{array}{rcll} \eta^1 F^2(g, k) \circ \eta^1 F^2(f, h) & = & F^1 C^2 k \circ F^2 g Y \circ F^2 f Y \circ F^1 A^2 h & \text{by (49)} \\ & = & F^1 C^2 k \circ (F^2 g \circ F^2 f) Y \circ F^1 A^2 h & \text{by (12)} \\ & = & F^1 C^2 k \circ F^2 (g \circ f) Y \circ F^1 A^2 h & \\ & = & F^1 C^2 k \circ F^1 C^2 h \circ F^2 (g \circ f) X & \text{by (16)} \\ & = & F^1 C^2 (k \circ h) \circ F^2 (g \circ f) X & \\ & = & \eta^1 F^2(g \circ f, k \circ h) & \text{by (49)} \\ & = & \eta^1 F^2((g, k) \circ (f, h)) & \end{array} \tag{53}$$

$$\begin{array}{rcll} \eta^1 F^2 1_{(A,X)} & = & \eta^1 F^2(1_A, 1_X) & \\ & = & F^2 1_A X \circ F^1 A^2 1_X & \text{by (49)} \\ & = & 1_{F^1 A} X \circ 1_{F^1 A^1 X} & \text{by (9)} \\ & = & 1_{F^1 A^1 X} \circ 1_{F^1 A^1 X} & \text{by (13)} \\ & = & 1_{\eta^1 F^1(A,X)} & \text{by (48).} \end{array} \tag{54}$$

By rule (10) with (48), (52), (53), and (54) as premises,

$$\eta^1 F \quad : \quad \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]. \tag{55}$$

Now we wish to show using (17) that $\eta^2\varphi : \mathsf{Fun}\,[\mathsf{C}\times\mathsf{D},\mathsf{E}]\,(\eta^1 F, \eta^1 G)$. For the typing, from (7) and (15), we have

$$\begin{array}{rcl} F^2 f & : & \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(F^1 A, F^1 B) \\ G^2 f & : & \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(G^1 A, G^1 B) \\ \varphi A & : & \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(F^1 A, G^1 A) \\ \varphi B & : & \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(F^1 B, G^1 B), \end{array}$$

therefore by (16),

$$\varphi B \circ F^2 f \quad = \quad G^2 f \circ \varphi A \;:\; \mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,(F^1 A, G^1 B). \tag{56}$$

Reasoning equationally to establish the third premise of (17),

$$\begin{array}{rcll} \eta^2\varphi(B,Y)\circ\eta^1 F^2(f,h) & = & \varphi BY \circ F^2 fY \circ F^1 A^2 h & \text{by (49) and (50)} \\ & = & (\varphi B \circ F^2 f)Y \circ F^1 A^2 h & \text{by (12)} \\ & = & G^1 B^2 h \circ (\varphi B \circ F^2 f)X & \text{by (16)} \\ & = & G^1 B^2 h \circ (G^2 f \circ \varphi A)X & \text{by (56)} \\ & = & G^1 B^2 h \circ G^2 fX \circ \varphi AX & \text{by (12)} \\ & = & \eta^1 G^2(f,h) \circ \eta^2\varphi(A,X) & \text{by (49) and (50).} \end{array}$$

This fact together with (55) and (51) establish all the premises of (17), therefore

$$\eta^2\varphi \quad : \quad \mathsf{Fun}\,[\mathsf{C}\times\mathsf{D},\mathsf{E}]\,(\eta^1 F, \eta^1 G). \tag{57}$$

Finally, reasoning equationally, we have

$$\begin{array}{rcll} \eta^2(\psi\circ\varphi)(A,X) & = & (\psi\circ\varphi)AX & \text{by (50)} \\ & = & \psi AX \circ \varphi AX & \text{by (12) twice} \\ & = & \eta^2\psi(A,X)\circ\eta^2\varphi(A,X) & \text{by (50)} \\ & = & (\eta^2\psi\circ\eta^2\varphi)(A,X) & \text{by (12),} \\ \eta^2 1_F(A,X) & = & 1_F AX & \text{by (50)} \\ & = & 1_{F^1 A^1 X} & \text{by (13) twice} \\ & = & 1_{\eta^1 F^1(A,X)} & \text{by (48)} \\ & = & 1_{\eta^1 F}(A,X) & \text{by (13).} \end{array}$$

Since $A : \mathsf{C}$ and $X : \mathsf{D}$ were arbitrary,

$$\eta^2(\psi\circ\varphi) \quad = \quad \eta^2\psi\circ\eta^2\varphi \tag{58}$$

$$\eta^2 1_F \quad = \quad 1_{\eta^1 F}. \tag{59}$$

By (10) using (55), (57), (58), and (59) as premises,

$$\eta \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C},\mathsf{Fun}\,[\mathsf{D},\mathsf{E}]\,]\,,\mathsf{Fun}\,[\mathsf{C}\times\mathsf{D},\mathsf{E}]\,]\,.$$

This establishes that η is a functor of the correct type.

Step 3 For this step, we work in the following type environment:

$$\begin{array}{rcl}
F, G & : & \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \\
H, K & : & \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,] \\
A, B & : & \mathsf{C} \\
X, Y & : & \mathsf{D} \\
f & : & \mathsf{C}(A, B) \\
h & : & \mathsf{D}(X, Y) \\
\varphi & : & \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(F, G) \\
\psi & : & \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,(H, K).
\end{array}$$

To show that θ^1 and η^1 are inverses, by (26)–(28) and (48)–(49),

$$\begin{array}{rcccl}
\eta^1(\theta^1 F)^1(A, X) & = & \theta^1 F^1 A^1 X & = & F^1(A, X), \\
\eta^1(\theta^1 F)^2(f, h) & = & \theta^1 F^2 f Y \circ \theta^1 F^1 A^2 h & = & F^2(f, h).
\end{array}$$

By extensionality, $\eta^1(\theta^1 F) = F$.

Also, by (26) and (48),

$$\theta^1(\eta^1 H)^1 A^1 X \quad = \quad \eta^1 H^1(A, X) \quad = \quad H^1 A^1 X,$$

and by (27), (49), (9), and (13),

$$\begin{array}{rcccccl}
\theta^1(\eta^1 H)^1 A^2 h & = & \eta^1 H^2(1_A, h) & = & H^1 A^2 h \circ H^2 1_A X & & \\
& = & H^1 A^2 h \circ 1_{H^1 A} X & = & H^1 A^2 h \circ 1_{H^1 A^1 X} & = & H^1 A^2 h.
\end{array}$$

Since $X : \mathsf{D}$ and $h : \mathsf{D}(X, Y)$ were arbitrary, $\theta^1(\eta^1 H)^1 A = H^1 A$, and since $A : \mathsf{C}$ was arbitrary, $\theta^1(\eta^1 H)^1 = H^1$. In addition, using (28), (49), and (9),

$$\begin{array}{rcccl}
\theta^1(\eta^1 H)^2 f X & = & \eta^1 H^2(f, 1_X) & = & H^2 f X \circ H^1 A^2 1_X \\
& = & H^2 f X \circ 1_{H^1 A^1 X} & = & H^2 f X,
\end{array}$$

and since $X : \mathsf{D}$ and $f : \mathsf{C}(A, B)$ were arbitrary, $\theta^1(\eta^1 H)^2 = H^2$, therefore $\theta^1(\eta^1 H) = H$.

To show that θ^2 and η^2 are inverses, by (29) and (50),

$$\begin{array}{rcccl}
\theta^2(\eta^2 \psi) A X & = & \eta^2 \psi(A, X) & = & \psi A X \\
\eta^2(\theta^2 \varphi)(A, X) & = & \theta^2 \varphi A X & = & \varphi(A, X).
\end{array}$$

Since $A : \mathsf{C}$ and $X : \mathsf{D}$ were arbitrary, $\theta^2(\eta^2 \psi) = \psi$ and $\eta^2(\theta^2 \varphi) = \varphi$.

Step 4 This step turns out to be the most involved of the four steps. For the remainder of the proof we will suppress detail in equational arguments, concentrating on the overall structure of the proof. All of the equational arguments are of the same flavor as those in steps 1–3 and are no more difficult.

Let Cat^3 abbreviate $\mathsf{Cat}^{op} \times \mathsf{Cat}^{op} \times \mathsf{Cat}$.

Recall that the symbols θ and η were used as abbreviations for $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$ in steps 1–3 above. Although the constructions and proofs of steps 1–3 depended on the parameters C, D, and E, they did so in a uniform way, so it was not necessary to mention the dependence explicitly. In a sense, the very fact that we were able to carry out steps 1–3 without reference to the particular nature of the categories C, D, and E is an indication that the construction was natural.

We would like to prove this formally. In this step, therefore, we will write $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$ for what was abbreviated as θ and η, respectively, in steps 1–3. The symbols θ and η will now take on their true meaning as natural transformations

$$
\begin{aligned}
\theta &: \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,(U, V) \\
\eta &: \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,(V, U),
\end{aligned}
$$

of which $\theta(\mathsf{CDE})$ and $\eta(\mathsf{CDE})$ are the components, for suitably defined functors $U, V : \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]$. We must derive this typing of θ and η to establish naturality.

For the first part of this step, we work in the following type environment:

$$
\begin{aligned}
P &: \mathsf{Fun}\,[\mathsf{L}, \mathsf{C}] = \mathsf{Cat}(\mathsf{L}, \mathsf{C}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{C}, \mathsf{L}) \\
Q &: \mathsf{Fun}\,[\mathsf{M}, \mathsf{D}] = \mathsf{Cat}(\mathsf{M}, \mathsf{D}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{D}, \mathsf{M}) \\
R &: \mathsf{Fun}\,[\mathsf{E}, \mathsf{N}] = \mathsf{Cat}(\mathsf{E}, \mathsf{N}) \\
I &: \mathsf{Fun}\,[\mathsf{P}, \mathsf{L}] = \mathsf{Cat}(\mathsf{P}, \mathsf{L}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{L}, \mathsf{P}) \\
J &: \mathsf{Fun}\,[\mathsf{Q}, \mathsf{M}] = \mathsf{Cat}(\mathsf{Q}, \mathsf{M}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{M}, \mathsf{Q}) \\
K &: \mathsf{Fun}\,[\mathsf{N}, \mathsf{R}] = \mathsf{Cat}(\mathsf{N}, \mathsf{R}) \\
F, G, H &: \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \\
\varphi &: \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(F, G) \\
\psi &: \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(G, H) \\
L, M, N &: \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,] \\
\sigma &: \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,(L, M) \\
\tau &: \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,(M, N) \\
A, B, C &: \mathsf{L} \\
X, Y, Z &: \mathsf{M} \\
f &: \mathsf{L}(A, B) \\
g &: \mathsf{L}(B, C) \\
h &: \mathsf{M}(X, Y) \\
k &: \mathsf{M}(Y, Z).
\end{aligned}
$$

First we define U and V and establish that they are of the correct type.

Define

$$
\begin{aligned}
U^1(\mathsf{CDE}) &\stackrel{\text{def}}{=} \mathsf{Fun}\,[\mathsf{C}\times\mathsf{D},\mathsf{E}] : \mathsf{Cat} && (60)\\
U^2(PQR)^1F^1(A,X) &\stackrel{\text{def}}{=} R^1(F^1(P^1A,Q^1X)) : \mathsf{N} && (61)\\
U^2(PQR)^1F^2(f,h) &\stackrel{\text{def}}{=} R^2(F^2(P^2f,Q^2h))\\
&\quad : \mathsf{N}(R^1(F^1(P^1A,Q^1X)),R^1(F^1(P^1B,Q^1Y)))\\
&\quad : \mathsf{N}(U^2(PQR)^1F^1(A,X),U^2(PQR)^1F^1(B,Y)) && (62)\\
U^2(PQR)^2\varphi(A,X) &\stackrel{\text{def}}{=} R^2(\varphi(P^1A,Q^1X))\\
&\quad : \mathsf{N}(R^1(F^1(P^1A,Q^1X)),R^1(G^1(P^1A,Q^1X)))\\
&\quad : \mathsf{N}(U^2(PQR)^1F^1(A,X),U^2(PQR)^1G^1(A,X)) && (63)
\end{aligned}
$$

The typing of the expressions on the right-hand sides of these equations follows from the analysis rules for functors and natural transformations and from substitution of (61).

By equational reasoning using (62), (2) and (8) for the first equation and (61), (62), (3), and (9) for the second,

$$
\begin{aligned}
&U^2(PQR)^1F^2((g,k)\circ(f,h))\\
&\qquad = U^2(PQR)^1F^2(g,k)\circ U^2(PQR)^1F^2(f,h) && (64)\\
&U^2(PQR)^1F^2\mathsf{1}_{(A,X)} = \mathsf{1}_{U^2(PQR)^1F^1(A,X)}. && (65)
\end{aligned}
$$

We can conclude from (10) using (61), (62), (64), and (65) as premises that

$$U^2(PQR)^1F \quad : \quad \mathsf{Fun}\,[\mathsf{L}\times\mathsf{M},\mathsf{N}] \tag{66}$$

and similarly

$$U^2(PQR)^1G \quad : \quad \mathsf{Fun}\,[\mathsf{L}\times\mathsf{M},\mathsf{N}] \tag{67}$$

From (16), we have

$$\varphi(P^1B,Q^1Y)\circ F^2(P^2f,Q^2h) \quad = \quad G^2(P^2f,Q^2h)\circ\varphi(P^1A,Q^1X).$$

By equational reasoning using this, (62), and (63),

$$
\begin{aligned}
&U^2(PQR)^2\varphi(B,Y)\circ U^2(PQR)^1F^2(f,h)\\
&\quad = U^2(PQR)^1G^2(f,h)\circ U^2(PQR)^2\varphi(A,X)
\end{aligned} \tag{68}
$$

By (17) with (66), (67), (63), and (68) as premises,

$$U^2(PQR)^2\varphi \quad : \quad \mathsf{Fun}\,[\mathsf{L}\times\mathsf{M},\mathsf{N}]\,(U^2(PQR)^1F,U^2(PQR)^1G). \tag{69}$$

Equational reasoning using (63), (61), (12), and (13) yields

$$
\begin{aligned}
U^2(PQR)^2(\psi\circ\varphi)(A,X) &= (U^2(PQR)^2\psi\circ U^2(PQR)^2\varphi)(A,X)\\
U^2(PQR)^2\mathsf{1}_F(A,X) &= \mathsf{1}_{U^2(PQR)^1F}(A,X),
\end{aligned}
$$

and since $A : \mathsf{L}$ and $X : \mathsf{M}$ were arbitrary,

$$\begin{aligned} U^2(PQR)^2(\psi \circ \varphi) &= U^2(PQR)^2\psi \circ U^2(PQR)^2\varphi && (70) \\ U^2(PQR)^2 1_F &= 1_{U^2(PQR)^1 F}. && (71) \end{aligned}$$

By (10) using (66), (69), (70), and (71) as premises,

$$U^2(PQR) \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}\,], \mathsf{Fun}\,[\mathsf{L} \times \mathsf{M}, \mathsf{N}\,]\,],$$

thus by (23) and (60),

$$U^2(PQR) \quad : \quad \mathsf{Cat}(U^1(\mathsf{CDE}), U^1(\mathsf{LMN})). \qquad (72)$$

The following equations can be established by purely equational reasoning:

$$\begin{aligned} U^2((IJK) \circ (PQR))^1 F^1(A, X) &= (U^2(IJK) \circ U^2(PQR))^1 F^1(A, X) \\ U^2((IJK) \circ (PQR))^1 F^2(f, h) &= (U^2(IJK) \circ U^2(PQR))^1 F^2(f, h) \\ U^2((IJK) \circ (PQR))^2 \varphi(A, X) &= (U^2(IJK) \circ U^2(PQR))^2 \varphi(A, X) \\ U^2 1^1_{(\mathsf{CDE})} F^1(A, X) &= 1^1_{U^1(\mathsf{CDE})} F^1(A, X) \\ U^2 1^1_{(\mathsf{CDE})} F^2(f, h) &= 1^1_{U^1(\mathsf{CDE})} F^2(f, h) \\ U^2 1^2_{(\mathsf{CDE})} \varphi(A, X) &= 1^2_{U^1(\mathsf{CDE})} \varphi(A, X). \end{aligned}$$

Since A, X, f, and h were arbitrary, by the extensionality rules,

$$\begin{aligned} U^2((IJK) \circ (PQR))^1 F^1 &= (U^2(IJK) \circ U^2(PQR))^1 F^1 \\ U^2((IJK) \circ (PQR))^1 F^2 &= (U^2(IJK) \circ U^2(PQR))^1 F^2 \\ U^2((IJK) \circ (PQR))^2 \varphi &= (U^2(IJK) \circ U^2(PQR))^2 \varphi \\ U^2 1^1_{(\mathsf{CDE})} F^1 &= 1^1_{U^1(\mathsf{CDE})} F^1 \\ U^2 1^1_{(\mathsf{CDE})} F^2 &= 1^1_{U^1(\mathsf{CDE})} F^2 \\ U^2 1^2_{(\mathsf{CDE})} \varphi &= 1^2_{U^1(\mathsf{CDE})} \varphi. \end{aligned}$$

By extensionality (20),

$$\begin{aligned} U^2((IJK) \circ (PQR))^1 F &= (U^2(IJK) \circ U^2(PQR))^1 F \\ U^2 1^1_{(\mathsf{CDE})} F &= 1^1_{U^1(\mathsf{CDE})} F, \end{aligned}$$

and since F and φ were arbitrary,

$$\begin{aligned} U^2((IJK) \circ (PQR))^1 &= (U^2(IJK) \circ U^2(PQR))^1 \\ U^2((IJK) \circ (PQR))^2 &= (U^2(IJK) \circ U^2(PQR))^2 \\ U^2 1^1_{(\mathsf{CDE})} &= 1^1_{U^1(\mathsf{CDE})} \\ U^2 1^2_{(\mathsf{CDE})} &= 1^2_{U^1(\mathsf{CDE})}. \end{aligned}$$

Again by (20),

$$\begin{aligned} U^2((IJK) \circ (PQR)) &= U^2(IJK) \circ U^2(PQR) && (73) \\ U^2 1_{(\mathsf{CDE})} &= 1_{U^1(\mathsf{CDE})}. && (74) \end{aligned}$$

By (10) using (60), (72), (73), and (74) as premises,

$$U \quad : \quad \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,. \tag{75}$$

This establishes the type of U.

The argument for V is similar, using the definitions

$$V^1(\mathsf{CDE}) \stackrel{\text{def}}{=} \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,] : \mathsf{Cat} \tag{76}$$

$$V^2(PQR)^1L^1A^1X \stackrel{\text{def}}{=} R^1(L^1(P^1A)^1(Q^1X)) : \mathsf{N} \tag{77}$$

$$\begin{aligned} V^2(PQR)^1L^1A^2h &\stackrel{\text{def}}{=} R^2(L^1(P^1A)^2(Q^2h)) \\ &: \mathsf{N}(R^1(L^1(P^1A)^1(Q^1X)), R^1(L^1(P^1A)^1(Q^1Y))) \\ &: \mathsf{N}(V^2(PQR)^1L^1A^1X, V^2(PQR)^1L^1A^1Y) \end{aligned} \tag{78}$$

$$\begin{aligned} V^2(PQR)^1L^2fX &\stackrel{\text{def}}{=} R^2(L^2(P^2f)(Q^1X)) \\ &: \mathsf{N}(R^1(L^1(P^1A)^1(Q^1X)), R^1(L^1(P^1B)^1(Q^1X))) \\ &: \mathsf{N}(V^2(PQR)^1L^1A^1X, V^2(PQR)^1L^1B^1X) \end{aligned} \tag{79}$$

$$\begin{aligned} V^2(PQR)^2\sigma AX &\stackrel{\text{def}}{=} R^2(\sigma(P^1A)(Q^1X)) \\ &: \mathsf{N}(R^1(L^1(P^1A)^1(Q^1X)), R^1(M^1(P^1A)^1(Q^1X))) \\ &: \mathsf{N}(V^2(PQR)^1L^1A^1X, V^2(PQR)^1M^1A^1X). \end{aligned} \tag{80}$$

Again, the typing of the expressions on the right-hand sides of these equations follows from the analysis rules for functors and natural transformations.

By three applications of (17), it follows from (77)–(80) that

$$V^2(PQR)^2\sigma A \quad : \quad \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,(V^2(PQR)^1L^1A, V^2(PQR)^1M^1A)$$

$$V^2(PQR)^2\sigma \quad : \quad \mathsf{Fun}\,[\mathsf{L}, \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,]\,(V^2(PQR)^1L, V^2(PQR)^1M) \tag{81}$$

$$V^2(PQR)^1L^2f \quad : \quad \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,(V^2(PQR)^1L^1A, V^2(PQR)^1L^1B) \tag{82}$$

From (10), using (77), (78), and the equations

$$\begin{aligned} V^2(PQR)^1L^1A^2(k \circ h) &= V^2(PQR)^1L^1A^2k \circ V^2(PQR)^1L^1A^2h \\ V^2(PQR)^1L^1A^2 1_X &= 1_{V^2(PQR)^1L^1A^1X} \end{aligned}$$

as premises, we obtain

$$V^2(PQR)^1L^1A : \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,. \tag{83}$$

Again from (10), using (83), (82), and the equations

$$\begin{aligned} V^2(PQR)^1L^2(g \circ f) &= V^2(PQR)^1L^2g \circ V^2(PQR)^1L^2f \\ V^2(PQR)^1L^2 1_A &= 1_{V^2(PQR)^1L^1A} \end{aligned}$$

as premises, we get

$$V^2(PQR)^1L \quad : \quad \mathsf{Fun}\,[\mathsf{L}, \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,]\,. \tag{84}$$

Still again from (10), using (84), (81), and the equations

$$\begin{aligned} V^2(PQR)^2(\tau \circ \sigma) &= V^2(PQR)^2\tau \circ V^2(PQR)^2\sigma \\ V^2(PQR)^2 1_L &= 1_{V^2(PQR)^1 L} \end{aligned}$$

as premises, we have

$$V^2(PQR) : \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,, \mathsf{Fun}\,[\mathsf{L}, \mathsf{Fun}\,[\mathsf{M}, \mathsf{N}]\,]\,]\,.$$

By (23) and (76),

$$V^2(PQR) : \mathsf{Cat}(V^1(\mathsf{CDE}), V^1(\mathsf{LMN})). \tag{85}$$

One concludes from (76) and (85) and the equations

$$\begin{aligned} V^2((IJK) \circ (PQR)) &= V^2(IJK) \circ V^2(PQR) \\ V^2 1_{(\mathsf{CDE})} &= 1_{V^1(\mathsf{CDE})} \end{aligned}$$

using (10) that

$$V \quad : \quad \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,. \tag{86}$$

This establishes the type of V.

For the last part of the proof, we wish to show that θ and η are natural transformations of the correct type. We have already shown in step 3 that they are inverses. For this part of the proof, we work in the type environment

$$\begin{aligned} P &: \mathsf{Fun}\,[\mathsf{L}, \mathsf{C}] = \mathsf{Cat}(\mathsf{L}, \mathsf{C}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{C}, \mathsf{L}) \\ Q &: \mathsf{Fun}\,[\mathsf{M}, \mathsf{D}] = \mathsf{Cat}(\mathsf{M}, \mathsf{D}) = \mathsf{Cat}^{\mathsf{op}}(\mathsf{D}, \mathsf{M}) \\ R &: \mathsf{Fun}\,[\mathsf{E}, \mathsf{N}] = \mathsf{Cat}(\mathsf{E}, \mathsf{N}) \\ F, G &: \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}] \\ \varphi &: \mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,(F, G) \\ A, B &: \mathsf{C} \\ X, Y &: \mathsf{D} \\ g &: \mathsf{C}(A, B) \\ h &: \mathsf{D}(X, Y). \end{aligned}$$

We showed in step 1 that

$$\theta(\mathsf{CDE}) \quad : \quad \mathsf{Fun}\,[\mathsf{Fun}\,[\mathsf{C} \times \mathsf{D}, \mathsf{E}]\,, \mathsf{Fun}\,[\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,]\,]\,.$$

It follows from (23), (60), and (76) that

$$\theta(\mathsf{CDE}) \quad : \quad \mathsf{Cat}(U^1(\mathsf{CDE}), V^1(\mathsf{CDE})), \tag{87}$$

and similarly

$$\theta(\mathsf{LMN}) \quad : \quad \mathsf{Cat}(U^1(\mathsf{LMN}), V^1(\mathsf{LMN})).$$

Combining these facts with (72) and (85) using (4),

$$\begin{aligned} \theta(\mathsf{LMN}) \circ U^2(PQR) &\quad:\quad \mathsf{Cat}(U^1(\mathsf{CDE}), V^1(\mathsf{LMN})) \\ V^2(PQR) \circ \theta(\mathsf{CDE}) &\quad:\quad \mathsf{Cat}(U^1(\mathsf{CDE}), V^1(\mathsf{LMN})). \end{aligned}$$

As part of the proof that θ is a natural transformation, we wish to show that these expression are equal.

By equational reasoning using the definitions (26)–(26), we can derive

$$\begin{aligned} (\theta(\mathsf{LMN}) \circ U^2(PQR))^1 F^1 A^1 X &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^1 F^1 A^1 X \\ (\theta(\mathsf{LMN}) \circ U^2(PQR))^1 F^1 A^2 h &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^1 F^1 A^2 h \\ (\theta(\mathsf{LMN}) \circ U^2(PQR))^1 F^2 g X &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^1 F^2 g X \\ (\theta(\mathsf{LMN}) \circ U^2(PQR))^2 \varphi A X &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^2 \varphi A X. \end{aligned}$$

Since h, g, X, and A were arbitrary,

$$\begin{aligned} (\theta(\mathsf{LMN}) \circ U^2(PQR))^1 &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^1 \\ (\theta(\mathsf{LMN}) \circ U^2(PQR))^2 &= (V^2(PQR) \circ \theta(\mathsf{CDE}))^2, \end{aligned}$$

thus by (20),

$$\theta(\mathsf{LMN}) \circ U^2(PQR) = V^2(PQR) \circ \theta(\mathsf{CDE}). \tag{88}$$

Now using (17) with (75), (86), (87), and (88) as premises,

$$\theta \quad : \quad \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,(U, V).$$

This establishes that θ is a natural transformation of the appropriate type.

The proof that $\eta : \mathsf{Fun}\,[\mathsf{Cat}^3, \mathsf{Cat}]\,(V, U)$ is similar. □

5 Conclusions and Future Work

In addition to Theorem 4.1, we have also developed a direct proof that Cat is cartesian closed. This involves establishing a particular adjunction, as illustrated in the following two diagrams:

$$\begin{tikzcd} \mathsf{C} \times \mathsf{D} \arrow[r, "F^2 H"] & \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}] \times \mathsf{D} \arrow[r, "\varepsilon"] & \mathsf{E} \arrow[d, "G^1"] \\ \mathsf{C} \arrow[u, "F^1"] \arrow[r, "H"] & \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}] \arrow[u, "F^1"] \arrow[r, "1"] & \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}] \end{tikzcd}$$

$$\begin{tikzcd} \mathsf{C} \times \mathsf{D} \arrow[r, "1"] & \mathsf{C} \times \mathsf{D} \arrow[r, "K"] \arrow[d, "G^1"] & \mathsf{E} \arrow[d, "G^1"] \\ \mathsf{C} \arrow[u, "F^1"] \arrow[r, "\eta"] & \mathsf{Fun}\,[\mathsf{D}, \mathsf{C} \times \mathsf{D}] \arrow[r, "G^2 K"] & \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}] \end{tikzcd}$$

Like Theorem 4.1, the proof breaks into several steps:

1. the definition and typing of the unit η of the adjuction,
2. the definition and typing of the counit ε of the adjuction,
3. the definition and typing of the left adjoint F,
4. the definition and typing of the right adjoint G,
5. the definition of a bijection between the two homsets $\mathsf{Cat}(\mathsf{C} \times \mathsf{D}, \mathsf{E})$ and $\mathsf{Cat}(\mathsf{C}, \mathsf{Fun}\,[\mathsf{D}, \mathsf{E}]\,)$ consisting of a pair of inverse maps $H \mapsto \varepsilon \circ F^2 H$ and $K \mapsto G^2 K \circ \eta$, and a proof that they are inverses.

The arguments are very similar to those in the proof of Theorem 4.1.

Several intriguing problems present themselves for future work. Of course, the most interesting prospect is the automation of the system. As one works with the system, it becomes quickly apparent that, as notationally complex as the proofs are, they can for the most part be developed in a purely mechanical fashion. Using backwards subgoaling starting from the desired conclusion and working backwards, the application of rules is largely syntax-directed and deterministic. Except for the equational arguments, very little thought is required; most of the work involves merely matching and substitution. The typing considerations alone dictate the overall structure of the proof, even determining to a large extent the definitions at the beginning of each step (e.g. (26)–(29) and (48)–(50)). Even the equational proofs tend to exhibit a nearly deterministic structure. This indicates strongly that most of the process can be fully automated, and the proof search can be made quite efficient.

We are currently investigating the possibility of implementing this system in the NuPrl automated deduction system [6]. NuPrl not only provides a general formalism for encoding proof rules, but it also provides a programming language for specifying tactics for automatic proof development.

Proofs in our system, as mentioned, tend to exhibit a discernable structure, at least for the verification of type judgements and to a lesser extent for the equations. Arguments tend to break down into the application of analysis rules followed by the application of synthesis rules, suggesting a normal form. Even the equational arguments tend to follow a certain structure, with the application of analysis rules followed by application of the extensionality rules. These observations point toward a normal form theorem.

Another interesting question is the computational complexity of the system. As mentioned, proofs tend to be largely syntax-directed and deterministic. This seems to indicate that the complexity of the system is low.

Acknowledgements

Thanks to Kamal Aboul-Hosn, Christoph Kreitz, Robert L. Constable, Neil Ghani, Frank J. Oles, Gordon Plotkin, and Glynn Winskel. This work was supported in part by NSF grant CCR-0105586 and by ONR Grant N00014-01-1-0968. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of these organizations or the US Government.

References

[1] A. Asperti and G. Longo. *Categories, Types and Structures: An introduction to category theory for the working computer scientist.* Foundations of Computing. MIT Press, 1991.

[2] Michael Barr and Charles Wells. *Category Theory for Computing Science.* Prentice Hall, 1990.

[3] I. Beylin and P. Dybjer. *Extracting a proof of coherence for monoidal categories from a proof of normalization for monoids*, volume 1158 of *Lecture Notes in Computer Science.* Springer-Verlag, 1996.

[4] M. J. Cáccamo, J. M. E. Hyland, and G. Winskel. Lecture notes in category theory. Technical Report LS-02-catnotes, BRICS, University of Aarhus, June 2002.

[5] M. J. Cáccamo and G. Winskel. A higher-order calculus for categories. Technical Report RS-01-27, BRICS, University of Aarhus, June 2001.

[6] Robert L. Constable, Stuart F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the Nuprl Development System.* Prentice-Hall, NJ, 1986.

[7] S. Eilenberg and S. MacLane. General theory of natural equivalences. *Trans. Amer. Math. Soc.*, 58:231–244, 1945.

[8] P. Eklund, M.A. Galán, J. Medina, M. Ojeda-Aciego, and A. Valverde. A graphical approach to monad compositions. *Electronic Notes in Theoretical Computer Science*, 40, 2002. http://www.elsevier.nl/locate/entcs/volume40.html.

[9] J. Goguen. A categorical manifesto. *Mathematical Structures in Computer Science*, 1(1):49–67, 1991.

[10] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, New York, 1971.

[11] Eugenio Moggi. Notions of computation and monads. *Inf. and Comp.*, 93, 1991.

[12] L. C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[13] G. M. Reed, A. W. Roscoe, and R. F. Wachter. *Topology and Category Theory in Computer Science*. Oxford University Press, 1991.

[14] D. Rydeheard and R. Burstall. *Computational Category Theory*. International Series in Computer Science. Prentice Hall, 1988.

[15] M. Takeyama. *Universal Structure and a Categorical Framework for Type Theory*. PhD thesis, University of Edinburgh, 1995.

[16] Philip Wadler. Monads for functional programming. In Johan Jeuring and Erik Meijer, editors, *Advanced Functional Programming: 1st Int. School on Advanced Functional Programming Techniques*, volume 925 of *Lecture Notes in Computer Science*, pages 24–52. Springer-Verlag, 1995.