A Higher-Order Calculus for Categories

Mario Jose Cáccamo
Glynn Winskel

BRICS Report Series RS-01-27

ISSN 0909-0878 June 2001

A Higher-Order Calculus for Categories

Mario Cáccamo Glynn Winskel

June, 2001

Abstract

A calculus for a fragment of category theory is presented. The types in the language denote categories and the expressions functors. The judgements of the calculus systematise categorical arguments such as: an expression is functorial in its free variables; two expressions are naturally isomorphic in their free variables. There are special binders for limits and more general ends. The rules for limits and ends support an algebraic manipulation of universal constructions as opposed to a more traditional diagrammatic approach. Duality within the calculus and applications in proving continuity are discussed with examples. The calculus gives a basis for mechanising a theory of categories in a generic theorem prover like Isabelle.

1 Introduction

A language for category theory [11] together with a calculus to derive natural isomorphisms are presented. We systematise categorical judgements of the kind: an expression is functorial in its free variables; there is an isomorphism between two expressions natural in their free variables. The resulting logic gives the foundations for a mechanisation of category theory in a generic theorem prover like Isabelle [15].

The language is based on the simply typed λ-calculus [2] where types denote categories and expressions functors between categories. We introduce constants, for hom-sets, for example, and binders for ends and coends. Ends are a generalisation of limits and play a central role in the calculus in increasing its expressive power.

The rules of the calculus enable a mechanical manipulation of formulae with ends and their dual coends. This gives a more calculational approach to categories than is usual, and less dependence on diagrams. This approach supplies tools for proving preservation of limits and colimits.

One motivation behind this work is the increasing use of the machinery of category theory in denotational semantics. There categories are often seen as generalised domains. An application of special interest to us is the use of presheaf categories as models for concurrency. A central result there is that functions between presheaf categories which preserve colimits, a form of continuity, preserve open-map bisimulation [9] (see [7] for details).

2 The Language

2.1 Categories as Types

In general, a category $\mathcal{C}$ consists of a class of objects and a class of arrows between any pair of objects A, B, written $\mathcal{C}(A, B)$, which support composition and have identities. If the class of arrows between any pair of objects is a set we say that $\mathcal{C}$ is *locally small* and call $\mathcal{C}(A, B)$ the hom-set for A, B. We will concentrate on locally small categories.

Locally small categories together with *functors*, arrows between categories, themselves form the large category **CAT**. Furthermore, given two locally small categories $\mathcal{C}$ and $\mathcal{D}$, the functors from $\mathcal{C}$ to $\mathcal{D}$ also form a category in which the objects are the functors and the arrows between them are *natural transformations*. We can summarise the rich structure of **CAT** in the diagram

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"{name=U}] \arrow[r, bend right, "G"'{name=D}] & \mathcal{D} \arrow[Rightarrow, from=U, to=D, "\theta"] \end{tikzcd}$$

where F and G are functors between locally small categories $\mathcal{C}$ and $\mathcal{D}$, and θ is a natural transformation from F to G.[1] We shall be particularly concerned with those natural transformations between functors which are isomorphisms, so-called *natural isomorphisms*.

The structure of **CAT** suggests a language. Its types will denote locally small categories; so we expect judgements which express the legitimate ways to build up types. Its expressions with free variables will

[1]In fact, making **CAT** an example of a 2-category.

denote functors from the category where the variables range to the category where the expression belongs; there will be judgements saying that expressions are functorial in their free variables. The diagram above suggests terms and judgements for constructing natural transformations. Here we will however restrict attention to just the judgements of there being a natural isomorphism between functors denoted by expressions. We plan to extend this work to a notation for natural transformations. Despite just considering natural isomorphisms, the calculus is surprisingly useful, allowing us to derive, for example, results on preservation of limits.

The constructors for types are interpreted as operations over categories. Given the categories $\mathcal{C}$ and $\mathcal{D}$,

- the opposite $\mathcal{C}^{\mathsf{op}}$ is the category whose objects are the objects of $\mathcal{C}$ and whose arrows are the arrows in $\mathcal{C}$ reversed, *i.e.*, $\mathcal{C}^{\mathsf{op}}(A, B) = \mathcal{C}(B, A)$ for any pair of objects A, B;

- the product $\mathcal{C} \times \mathcal{D}$ is the category whose objects and arrows are ordered pairs of objects and arrows in $\mathcal{C}$ and $\mathcal{D}$;

- the sum $\mathcal{C} + \mathcal{D}$ is the category whose objects and arrows are given by the disjoint union of objects and arrows in $\mathcal{C}$ and $\mathcal{D}$; and

- the functor category $[\mathcal{C}, \mathcal{D}]$ is the category of functors from $\mathcal{C}$ to $\mathcal{D}$ and the natural transformation between them.

Observe that a functor category built out of locally small categories is not necessarily locally small (Crole discusses this point in [8, page 61]). We constrain the use of functor categories $[\mathcal{C}, \mathcal{D}]$ to the case where $\mathcal{C}$ is small. A locally small category is *small* when the collection of objects is a set. Simple examples of small categories are $\mathbf{0}$ and $\mathbf{1}$, the empty and singleton categories respectively. We write $\mathbb{C}, \mathbb{D}, \mathbb{I}, \mathbb{J}, \cdots$ for small categories.

The syntax for types as locally small categories is

$$\mathcal{C} ::= \mathbf{Set} \mid \mathbf{0} \mid \mathbf{1} \mid \mathcal{C}^{\mathsf{op}} \mid \mathcal{C}_1 \times \mathcal{C}_2 \mid \mathcal{C}_1 + \mathcal{C}_2 \mid [\mathbb{C}, \mathcal{C}] \ .$$

Thus, for the language *locally small category* and *type* are synonymous.

2.2 Syntax for Expressions

The expressions $E_1, E_2, \cdots$ are defined by

$$\begin{aligned}E ::=\,& X \mid 1 \mid \lambda X^{\mathbb{C}}.E \mid E_1(E_2) \mid \mathcal{C}(E_1, E_2) \mid (E_1, E_2) \mid \\ & \mathit{fst}(E) \mid \mathit{snd}(E) \mid \mathit{inl}(E) \mid \mathit{inr}(E) \mid \mathit{case}^{\mathcal{C}+\mathcal{D}}(E_1, E_2, E_3) \mid \\ & E_1 \times E_2 \mid E_1 + E_2 \mid \int_{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} E \mid \int^{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} E\end{aligned}$$

where $\mathcal{C}$, $\mathcal{D}$ and $\mathbb{C}$ are types, and X is drawn from a countably infinite set of variables. The constant 1 stands for the singleton set.

The syntax is that of the simply typed λ-calculus à la Church with products and sums. To simplify the notation we sometimes ommit the type annotations in the lambda abstractions. The "integral" binders denote ends and coends. Ends extend the concept of limit to functors of mixed variance with domain of the form $\mathbb{C}^{\mathsf{op}} \times \mathbb{C}$. The treatment of ends given in this work supports a mechanical manipulation of these binders whose properties resemble the properties of the integral in mathematical analysis. Ends are discussed later – they are a central notion of the language for categories.

2.3 Functoriality

Not all possible expressions in the language give rise to functors. An example of a non-functorial expression is $\mathcal{C}(X, X)$ for a nontrivial category $\mathcal{C}$; there is no action over arrows matching the action over objects. Well-typed expressions in the language represent those expressions functorial in their free variables.

The syntactic judgement $X_1 : \mathcal{C}_1, \cdots, X_n : \mathcal{C}_n \vdash E : \mathcal{C}$ says that the expression E of type $\mathcal{C}$ is functorial in the free variables occurring in the context $X_1 : \mathcal{C}_1, \cdots, X_n : \mathcal{C}_n$; *i.e.* it denotes a functor from $\mathcal{C}_1 \times \ldots \times \mathcal{C}_n$ to $\mathcal{C}$. A context is a finite sequence of variable declarations. In informal mathematical usage one says $E(X_1, \cdots, X_n)$ is functorial in $X_1, \cdots, X_n$. Thus, an expression-in-context has two possible readings: as an object when the free variables are interpreted as objects, and as an arrow when the free variables are interpreted as arrows.

From the typing rules we derive well-typed expressions enforcing functoriality. One example is the rule for hom-expressions:

$$\mathsf{hom}\ \frac{\Gamma \vdash E_1 : \mathcal{C} \qquad \Gamma' \vdash E_2 : \mathcal{C}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash \mathcal{C}(E_1, E_2) : \mathbf{Set}}$$

where Γ and Γ' are contexts[2] and Γ^{op} is obtained from Γ by replacing each occurrence of a type $\mathcal{C}$ by $\mathcal{C}^{\mathsf{op}}$. We insist, also in the syntax, that $(\mathcal{C}^{\mathsf{op}})^{\mathsf{op}} = \mathcal{C}$ and that forming the opposite category respects the type constructions so *e.g.* $(\mathcal{C} \times \mathcal{D})^{\mathsf{op}} = \mathcal{C}^{\mathsf{op}} \times \mathcal{D}^{\mathsf{op}}$ and $[\mathcal{C}, \mathcal{D}]^{\mathsf{op}} = [\mathcal{C}^{\mathsf{op}}, \mathcal{D}^{\mathsf{op}}]$.

The rule for lambda abstraction introduces functor categories:

$$\mathsf{lam}\ \frac{\Gamma, X : \mathbb{C} \vdash E : \mathcal{D}}{\Gamma \vdash \lambda X^{\mathbb{C}}.E : [\mathbb{C}, \mathcal{D}]}$$

with the assumption that $\mathbb{C}$ is small. If E is an expression with free variables W, Z, we abbreviate $\lambda X^{\mathbb{C}\times\mathbb{D}}.E[\mathit{fst}(X)/W, \mathit{snd}(X)/Z]$ as $\lambda W^{\mathbb{C}}, Z^{\mathbb{D}}.E$.

There is a symmetric rule for eliminating functor categories:

$$\mathsf{app}\ \frac{\Gamma \vdash F : [\mathbb{C}, \mathcal{D}] \quad \Gamma' \vdash E : \mathbb{C}}{\Gamma, \Gamma' \vdash F(E) : \mathcal{D}.}$$

The derivation below shows $X : \mathbb{C} \vdash \lambda Y^{\mathbb{C}^{\mathsf{op}}}.\mathbb{C}(Y, X) : [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ which denotes the so-called Yoneda functor for $\mathbb{C}$:

$$\dfrac{\dfrac{\dfrac{}{Y : \mathbb{C} \vdash Y : \mathbb{C}}\,\mathsf{ass} \qquad \dfrac{}{X : \mathbb{C} \vdash X : \mathbb{C}}\,\mathsf{ass}}{X : \mathbb{C}, Y : \mathbb{C}^{\mathsf{op}} \vdash \mathbb{C}(Y, X) : \mathbf{Set}}\,\mathsf{hom} + \mathsf{exc}}{X : \mathbb{C} \vdash \lambda Y^{\mathbb{C}^{\mathsf{op}}}.\mathbb{C}(Y, X) : [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]}\,\mathsf{lam}$$

where the rule (exc) allows us to permute the variables in the contexts (there are rules for weakening and contraction as well).

2.4 Naturality

Given two expressions $E_1(X_1, \cdots, X_n)$ and $E_2(X_1, \cdots, X_n)$ functorial in the variables $X_1, \cdots, X_n$ it is sensible to ask whether the associated functors are naturally isomorphic. When they are, one says

$$E_1(X_1, \cdots, X_n) \cong E_2(X_1, \cdots, X_n)$$

natural in $X_1, \cdots, X_n$. More formally, the syntactic judgement $\Gamma \vdash E_1 \cong E_2 : \mathcal{C}$ says the expressions E_1 and E_2, where $\Gamma \vdash E_1 : \mathcal{C}$ and $\Gamma \vdash E_2 : \mathcal{C}$, are naturally isomorphic in the variables of Γ. The judgement $\Gamma \vdash E_1 \cong$

[2]The variables that appear in a context are distinct and different contexts have disjoint set of variables; any conflict is avoided by renaming.

$E_2 : \mathcal{C}$ asserts the existence of a natural isomorphism between the functors denoted by E_1 and E_2.

The relation defined by isomorphism is an equivalence, and there are rules for reflexivity, symmetry and transitivity. The rules for isomorphisms encode useful facts in category theory. There are, for example, rules for the Yoneda lemma and its corollary:

Theorem 2.1 (Yoneda lemma) *Let $\mathcal{C}$ be a locally small category. Then*

$$[\mathcal{C}^{\mathsf{op}}, \mathbf{Set}]\big(\lambda X.\mathcal{C}(X, C), F\big) \cong F(C)$$

natural in $C \in \mathcal{C}$ and $F \in [\mathcal{C}^{\mathsf{op}}, \mathbf{Set}]$.

A special case of the Yoneda lemma is expressed by the rule

$$\mathsf{yon}\ \frac{\Gamma, X : \mathbb{C}^{\mathsf{op}} \vdash E : \mathbf{Set}}{\Gamma, Z : \mathbb{C}^{\mathsf{op}} \vdash E[Z/X] \cong [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}](\lambda X^{\mathbb{C}^{\mathsf{op}}}.\mathbb{C}(X, Z), \lambda X^{\mathbb{C}^{\mathsf{op}}}.E) : \mathbf{Set}}\ .$$

That the Yoneda functor is full and faithful follows by replacing F in Theorem 2.1 with the functor $\lambda X.\mathcal{C}(X, D)$ for some D in $\mathcal{C}$. This, together with the fact that full and faithful functors preserve isomorphisms, gives:

Corollary 2.2 *$C \cong D$ iff $\lambda X.\mathcal{C}(X, C) \cong \lambda X.\mathcal{C}(X, D)$ for $C, D \in \mathcal{C}$.*

This is encoded in the calculus by the rule:

$$\mathsf{rep}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}} \vdash \mathcal{C}(X, E_1) \cong \mathcal{C}(X, E_2) : \mathbf{Set}}{\Gamma \vdash E_1 \cong E_2 : \mathcal{C}}$$

with the assumption that X is not free in E_1 or E_2. A complete presentation of the rules is postponed until section 4.

3 Ends and Coends

3.1 Representability

The manipulation of ends relies on the theory of representable functors.

Definition 3.1 (Representable Functor) *A functor $F : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$ is representable if for some object C in $\mathcal{C}$ there is an isomorphism*

$$\mathcal{C}(X, C) \cong F(X)$$

natural in X.

A *representation* for $F : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$ is a pair (C, θ) such that

$$\mathcal{C}(X, C) \overset{\theta_X}{\cong} F(X)$$

natural in X. A trivial example of a representable functor is $\lambda X.\mathcal{C}(X, A)$ for some object A in $\mathcal{C}$. There is a dual definition for $G : \mathcal{C} \to \mathbf{Set}$: a representation for G is a pair (D, ψ) such that

$$\mathcal{C}(D, X) \overset{\psi_X}{\cong} G(D)$$

natural in X. Below we see that limits and more generally ends are representations for special functors. The next result is a consequence of the Yoneda functor being full and faithful (see [6] for a proof):

Theorem 3.2 (Parametrised Representability) *Let $F : \mathcal{A} \times \mathcal{B}^{\mathsf{op}} \to$ $\mathbf{Set}$ be a functor such that for every object A in $\mathcal{A}$ there exists a representation $(\mathcal{B}(A), \theta^A)$ for the functor $\lambda X.F(A, X)$, then there is a unique extension of the mapping $A \mapsto \mathcal{B}(A)$ to a functor $\lambda X.\mathcal{B}(X) : \mathcal{A} \to \mathcal{B}$ such that*

$$\mathcal{B}(X, \mathcal{B}(A)) \overset{(\theta^A)_X}{\cong} F(A, X)$$

is natural in $A \in \mathcal{A}$ and $X \in \mathcal{B}^{\mathsf{op}}$.

This result shows that representations are functorial in their parameters; this will be crucial in justifying the typing rules for end formulae.

3.2 Limits and Ends as Representations

Definition 3.3 (Limit) *A limit of a functor $F : \mathbb{I} \to \mathcal{C}$ is a representation (L, θ) for $\lambda X.[\mathbb{I}, \mathcal{C}](\lambda Y.X, F) : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$, i.e.,*

$$\mathcal{C}(X, L) \overset{\theta_X}{\cong} [\mathbb{I}, \mathcal{C}](\lambda Y.X, F) \tag{1}$$

natural in $X \in \mathcal{C}^{\mathsf{op}}$.

The object L in (1) is often written in the literature as $\varprojlim_{\mathbb{I}} F$ and called the limit of F. The right hand side of (1) is the set of natural transformations from the constant functor $\lambda Y.X$ to F. Given $\gamma \in [\mathbb{I}, \mathcal{C}](\lambda Y.X, F)$

and an arrow $u : I \to J$ in $\mathbb{I}$ the naturality condition ensures that the diagram

$$\begin{tikzcd}
 & F(I) \arrow[dd, "F(u)"] \\
X \arrow[ur, "\gamma_I"] \arrow[dr, "\gamma_J"] & \\
 & F(J)
\end{tikzcd}$$

commutes. A natural transformation of this kind is a *cone* from X to F. The isomorphism θ_X in (1) establishes a one-to-one correspondence between arrows from X to L and cones from X to F. From the naturality condition on X we recover the more concrete definition of limits where $\theta_L(\mathsf{id}_L)$ is the universal cone. Dually, a *colimit* for $F : \mathbb{I} \to \mathcal{C}$ is a representation for the functor $\lambda X.[\mathbb{I}, \mathcal{C}](F, \lambda Y.X) : \mathcal{C} \to \mathbf{Set}$.

The notion of end extends that of limit to functors of mixed-variance, *e.g.* with domain $\mathbb{I}^{\mathsf{op}} \times \mathbb{I}$. In this setting, cones are generalised to wedges. Given a functor $G : \mathbb{I}^{\mathsf{op}} \times \mathbb{I} \to \mathcal{D}$, a *wedge* β from X to G is a family of arrows $\langle \beta_I : X \to G(I, I) \rangle_{I \in \mathbb{I}}$ in $\mathcal{D}$ such that for any arrow $u : I \to J$ in $\mathbb{I}$ the diagram

$$\begin{tikzcd}
 & G(I,I) \arrow[dr, "G(\mathsf{id}_I{,}u)"] & \\
X \arrow[ur, "\beta_I"] \arrow[dr, "\beta_J"'] & & G(I,J) \\
 & G(J,J) \arrow[ur, "G(u{,}\mathsf{id}_J)"'] &
\end{tikzcd}$$

commutes. Just as cones are natural transformations, so are wedges dinatural transformations – see [11, page 218] for the definition of dinatural transformations. Dinatural transformations do not compose in general, but they do compose with natural transformations. So there is a functor

$$\lambda F, G.\mathbf{Dinat}\big(F, G\big) : [\mathbb{I}^{\mathsf{op}} \times \mathbb{I}, \mathcal{D}]^{\mathsf{op}} \times [\mathbb{I}^{\mathsf{op}} \times \mathbb{I}, \mathcal{D}] \to \mathbf{Set}$$

where $\mathbf{Dinat}\big(F, G\big)$ is the set of dinatural transformations from F to G. A wedge from X to G is an element in $\mathbf{Dinat}\big(\lambda Y.X, G\big)$.

Definition 3.4 (End) *An end of a functor $G : \mathbb{I}^{\mathsf{op}} \times \mathbb{I} \to \mathcal{D}$ is a representation (E, θ) for $\lambda X.\mathbf{Dinat}\big(\lambda Y.X, G\big) : \mathcal{D}^{\mathsf{op}} \to \mathbf{Set}$, i.e.,*

$$\mathcal{D}(X, E) \overset{\theta_X}{\cong} \mathbf{Dinat}\big(\lambda Y.X, G\big) \tag{2}$$

natural in $X \in \mathcal{D}^{\mathsf{op}}$.

Following a similar analysis to that with limits, we may recover a more concrete definition for ends where $\theta_E(\mathsf{id}_E)$ is the universal wedge. In the language for categories the object E in (2) is written as $\int_{X^{\mathbb{I}^{\mathsf{op}}},Y^{\mathbb{I}}} G(X,Y)$, and more informally, and economically, as $\int_{Z^{\mathbb{I}}} G(Z^-,Z^+)$ where $Z^- : \mathbb{I}^{\mathsf{op}}$ and $Z^+ : \mathbb{I}$.

Dually, a *coend* of $G : \mathbb{I}^{\mathsf{op}} \times \mathbb{I} \to \mathcal{C}$ is a representation for the functor $\lambda X.\mathbf{Dinat}\big(G, \lambda Y.X\big)$. We write $\int^{X^{\mathbb{I}^{\mathsf{op}}},Y^{\mathbb{I}}} G(X,Y)$ for the coend of G (more informally, $\int^{Z^{\mathbb{I}}} G(Z^-,Z^+)$).

3.3 Ends with Parameters

A special case of parametrised representability arises when considering ends as representations. Suppose the functor $F : \mathcal{A}\times\mathbb{I}^{\mathsf{op}}\times\mathbb{I} \to \mathcal{B}$ such that for any object A in $\mathcal{A}$ the induced functor $\lambda X,Y.F(A,X,Y) : \mathbb{I}^{\mathsf{op}}\times\mathbb{I} \to \mathcal{B}$ has as end the representation $(\int_{Z^{\mathbb{I}}} F(A,Z^-,Z^+), \theta^A)$. Then the mapping $A \mapsto \int_{Z^{\mathbb{I}}} F(A,Z^-,Z^+)$ extends uniquely to a functor from $\mathcal{A}$ to $\mathcal{B}$ such that

$$\mathcal{B}(X, \textstyle\int_{Z^{\mathbb{I}}} F(A,Z^-,Z^+)) \overset{(\theta^A)_X}{\cong} [\mathbb{I}^{\mathsf{op}}\times\mathbb{I},\mathcal{B}]\big((\lambda Y,W.X),(\lambda Y,W.F(A,Y,W))\big)$$

natural in $A \in \mathcal{A}$ and $X \in \mathcal{B}^{\mathsf{op}}$. This is just Theorem 3.2 applied to the functor $\lambda A,X.[\mathbb{I}^{\mathsf{op}}\times\mathbb{I},\mathcal{B}]\big((\lambda Y,W.X),(\lambda Y,W.F(A,Y,W))\big)$. We conclude that the process of taking ends does not disturb functoriality over the variables which remain free. This justifies the rule for ends:

$$\mathsf{int}\ \frac{\Gamma, X:\mathbb{C}^{\mathsf{op}}, Y:\mathbb{C} \vdash E:\mathcal{D}}{\Gamma \vdash \int_{X^{\mathbb{C}^{\mathsf{op}}},Y^{\mathbb{C}}} E:\mathcal{D}}$$

where $\mathcal{D}$ is complete, *i.e.*, $\mathcal{D}$ has all ends (and equivalently, all limits).

Limits correspond to ends where the functor is extended with a dummy argument.[3] Thus, by using the rules (int), weakening (wea) and exchange (exc) we can form the derivation:

$$\cfrac{\cfrac{\Gamma, Y:\mathbb{C} \vdash E:\mathcal{D}}{\Gamma, X:\mathbb{C}^{\mathsf{op}}, Y:\mathbb{C} \vdash E:\mathcal{D}}\ \mathsf{wea} + \mathsf{exc}}{\Gamma \vdash \int_{X^{\mathbb{C}^{\mathsf{op}}},Y^{\mathbb{C}}} E:\mathcal{D}\ .}\ \mathsf{int}$$

[3]Conversely, an end can be regarded as a limit: given a functor $F : \mathbb{I}^{\mathsf{op}} \times \mathbb{I} \to \mathcal{C}$ there is a category $\mathbb{I}^{\S}$ together with a functor $d^{\S} : \mathbb{I}^{\S} \to \mathbb{I}^{\mathsf{op}}\times\mathbb{I}$ such that $\int_{I^{\mathbb{I}}} F(I^-,I^+) \cong \varprojlim_{\mathbb{I}}(F\circ d^{\S})$; for the details of this construction see [6, 11].

As the variable X is not free in the expression E the last judgement might be abbreviated as just $\Gamma \vdash \int_{Y^{\mathbb{C}}} E : \mathcal{D}$. Hence, we can use the integral for both ends and limits.

3.4 Complete Categories

We restrict the application of rules for ends to complete categories. A category is complete when it has all limits. The category **Set**, for example, is complete.

The set of natural transformations from F to G is characterised by an end expression, the so-called *naturality formula*:

$$[\mathbb{I}, \mathcal{D}](F, G) \cong \int_{I^{\mathbb{I}}} \mathcal{D}\big(F(I^-), G(I^+)\big)$$

natural in $F \in [\mathbb{I}, \mathcal{D}]^{\mathsf{op}}$ and $G \in [\mathbb{I}, \mathcal{D}]$. This isomorphism is explained by giving a concrete choice for the end in **Set** (see [6, 11] for details). In the calculus there is a rule for this formula:

$$\mathsf{nat}\ \frac{\Gamma, X : \mathbb{C} \vdash E_1 : \mathcal{D} \qquad \Gamma', Y : \mathbb{C} \vdash E_2 : \mathcal{D}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash [\mathbb{C}, \mathcal{D}](\lambda X^{\mathbb{C}}.E_1, \lambda Y^{\mathbb{C}}.E_2) \cong \int_{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} \mathcal{D}\big(E_1, E_2\big) : \mathbf{Set}}\ .$$

Similarly, there is an end expression for dinatural transformations:

$$\mathbf{Dinat}\big(F, G\big) \cong \int_{I^{\mathbb{I}}} \mathcal{D}\big(F(I^+, I^-), G(I^-, I^+)\big) \tag{3}$$

natural in $F \in [\mathbb{I}^{\mathsf{op}} \times \mathbb{I}, \mathcal{D}]^{\mathsf{op}}$ and $G \in [\mathbb{I}^{\mathsf{op}} \times \mathbb{I}, \mathcal{D}]$. By composing the isomorphism for the definition of ends (2) with an instance of the *dinaturality formula* (3) where F is a constant functor, we obtain the isomorphism:

$$\mathcal{D}\big(X, \int_{I^{\mathbb{I}}} G(I^-, I^+)\big) \cong \int_{I^{\mathbb{I}}} \mathcal{D}\big(X, G(I^-, I^+)\big) \tag{4}$$

natural in $X \in \mathcal{D}^{\mathsf{op}}$ and $G \in [\mathbb{I}^{\mathsf{op}} \times \mathbb{I}, \mathcal{D}]$. This formula shows how ends can be moved outside of a hom-expression. To avoid the introduction of special syntax for dinaturals in the language for categories we adopt (4) as the definition for ends:

$$\mathsf{end}\ \frac{\Gamma, X : \mathbb{C}^{\mathsf{op}}, Y : \mathbb{C} \vdash E : \mathcal{D}}{\Gamma, W : \mathcal{D}^{\mathsf{op}} \vdash \mathcal{D}\big(W, \int_{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} E\big) \cong \int_{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} \mathcal{D}(W, E) : \mathbf{Set}}$$

where $\mathcal{D}$ is complete.

The definition of limits is derivable from (end), (nat) and weakening (wea). First we derive a special case of the naturality formula:

$$\dfrac{\dfrac{\dfrac{}{W:\mathcal{D}\vdash W:\mathcal{D}}\,\textsf{ass}}{W:\mathcal{D},X:\mathbb{C}\vdash W:\mathcal{D}}\,\textsf{wea} \qquad \begin{matrix}\vdots\\ \Gamma,Y:\mathbb{C}\vdash E:\mathcal{D}\end{matrix}}{\Gamma,W:\mathcal{D}^{\mathsf{op}}\vdash [\mathbb{C},\mathcal{D}](\lambda X^{\mathbb{C}}.W,\lambda Y^{\mathbb{C}}.E)\cong\int_{X^{\mathbb{C}^{\mathsf{op}}},Y^{\mathbb{C}}}\mathcal{D}(W,E):\mathbf{Set}\ .}\,\textsf{nat + exc}$$

This is combined by means of transitivity of isomorphism with

$$\Gamma,W:\mathcal{D}^{\mathsf{op}}\vdash \mathcal{D}(W,\textstyle\int_{X^{\mathbb{C}^{\mathsf{op}}},Y^{\mathbb{C}}}E)\cong\int_{X^{\mathbb{C}^{\mathsf{op}}},Y^{\mathbb{C}}}\mathcal{D}(W,E):\mathbf{Set}\ .$$

Finally (with some rewriting):

$$\Gamma,W:\mathcal{D}^{\mathsf{op}}\vdash \mathcal{D}(W,\textstyle\int_{Y^{\mathbb{C}}}E)\cong[\mathbb{C},\mathcal{D}](\lambda X^{\mathbb{C}}.W,\lambda X^{\mathbb{C}}.E):\mathbf{Set}\ .$$

An important result in reasoning about ends is the *Fubini theorem*:

$$\textsf{fub}\ \dfrac{\Gamma,X:\mathbb{I}^{\mathsf{op}},Y:\mathbb{I},W:\mathbb{J}^{\mathsf{op}},Z:\mathbb{J}\vdash E:\mathcal{D}}{\Gamma\vdash\int_{X^{\mathbb{I}^{\mathsf{op}}},Y^{\mathbb{I}}}\int_{W^{\mathbb{J}^{\mathsf{op}}},Z^{\mathbb{J}}}E\cong\int_{W^{\mathbb{J}^{\mathsf{op}}},Z^{\mathbb{J}}}\int_{X^{\mathbb{I}^{\mathsf{op}}},Y^{\mathbb{I}}}E:\mathcal{D}}$$

where $\mathcal{D}$ is complete.

3.5 Duality: Coend Formulae

In **CAT**, any functor $F:\mathcal{C}\to\mathcal{D}$ is mirrored by its dual, a functor $F^*:\mathcal{C}^{\mathsf{op}}\to\mathcal{D}^{\mathsf{op}}$ which acts, as a function, in exactly the same way as F on objects and arrows. This dual view also involves natural transformations and is given by applying the 2-functor $(-)^*:\mathbf{CAT}\to\mathbf{CAT}$, which acts

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left=50, "F"{name=U}] \arrow[r, bend right=50, "G"'{name=D}] \arrow[Rightarrow, from=U, to=D, "\theta"] & \mathcal{D} \arrow[r, mapsto, shorten=1em] & \mathcal{C}^{\mathsf{op}} \arrow[r, bend left=50, "F^*"{name=U2}] \arrow[r, bend right=50, "G^*"'{name=D2}] \arrow[Rightarrow, from=D2, to=U2, "\theta^*"'] & \mathcal{D}^{\mathsf{op}}
\end{tikzcd}\ ,$$

where the components of θ^* are opposites of the components of θ. Note that dualising twice gives the identity. (Although $(-)^*$ reverses natural transformations, this does not have a direct effect in the calculus since we are only concerned with natural isomorphisms.)

Like a mirror, dualising affects our view of things and the way we describe them. A judgement $\Gamma\vdash E:\mathcal{D}$ denotes a functor whose dual is described by a dual form of judgement, $\Gamma^{\mathsf{op}}\vdash E^*:\mathcal{D}^{\mathsf{op}}$, where E^* is the expression obtained by turning ends into coends and coends into ends in E, adjusting the types of the bound variables. For example, the dual form of $Z:\mathcal{C}\vdash\int_{X^{\mathbb{I}^{\mathsf{op}}},Y^{\mathbb{I}}}E:\mathcal{D}$ is $Z:\mathcal{C}^{\mathsf{op}}\vdash\int^{Y^{\mathbb{I}^{\mathsf{op}}},X^{\mathbb{I}}}E^*:\mathcal{D}^{\mathsf{op}}$ where

E is an expression with free variables amongst X, Y, Z. The dual form of a product $E_1 \times E_2$ in **Set** is the sum $E_1^* + E_2^*$ in $\mathbf{Set}^{\mathsf{op}}$. It follows that $(E^*)^* = E$. In a similar way we can dualise judgements about the existence of natural isomorphisms, and so embody dualisation in the rules:

$$\mathsf{dua}\,\frac{\Gamma \vdash E : \mathcal{D}}{\Gamma^{\mathsf{op}} \vdash E^* : \mathcal{D}^{\mathsf{op}}} \qquad \mathsf{dual}\,\frac{\Gamma \vdash E_1 \cong E_2 : \mathcal{D}}{\Gamma^{\mathsf{op}} \vdash E_1^* \cong E_2^* : \mathcal{D}^{\mathsf{op}}.}$$

We can now, for example, derive the rule ($\mathsf{int*}$) for typing coends:

$$\cfrac{\cfrac{\cfrac{\Gamma, X : \mathbb{C}^{\mathsf{op}}, Y : \mathbb{C} \vdash E : \mathcal{D}}{\Gamma^{\mathsf{op}}, Y : \mathbb{C}^{\mathsf{op}}, X : \mathbb{C} \vdash E^* : \mathcal{D}^{\mathsf{op}}}\ \mathsf{dua} + \mathsf{exc}}{\Gamma^{\mathsf{op}} \vdash \int_{Y^{\mathbb{C}^{\mathsf{op}}}, X^{\mathbb{C}}} E^* : \mathcal{D}^{\mathsf{op}}}\ \mathsf{int}}{\Gamma \vdash \int^{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} E : \mathcal{D}.}\ \mathsf{dua}$$

A judgement for $\Gamma^{\mathsf{op}}, \Gamma' \vdash \mathcal{C}(E_1, E_2) : \mathbf{Set}$ where $\Gamma \vdash E_1 : \mathcal{C}$ and $\Gamma' \vdash E_2 : \mathcal{C}$ denotes the composition of the functor $\mathcal{C}(-,+) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$ with the functors $E_1^* : \Gamma^{\mathsf{op}} \to \mathcal{C}^{\mathsf{op}}$ and $E_2 : \Gamma \to \mathcal{C}$. Notice that, in keeping with practice, an expression occurring on the left of a hom-expression is implicitly dualised. This affects the definition of substitution of expressions for free variables.

The substitution $E_1[E_2/X]$ replaces the negative occurrences of the free variable X (on the left in an odd number of hom-expressions) by E_2^* and other occurrences by E_2. Formally, a substitution $E_1[E_2/X]$ is defined by induction on the structure of E_1 where the defining clause for hom-expressions is

$$\mathcal{C}(E', E'')[E_2/X] = \mathcal{C}(E'[E_2^*/X], E''[E_2/X]).$$

Because $\mathcal{C}(X, Y) \cong \mathcal{C}^{\mathsf{op}}(Y, X)$, for $X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{C}$, by substitution and dualisation we obtain the rule

$$\mathsf{opp}\,\frac{\Gamma \vdash E_1 : \mathcal{C} \quad \Gamma' \vdash E_2 : \mathcal{C}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash \mathcal{C}(E_1, E_2) \cong \mathcal{C}^{\mathsf{op}}(E_2^*, E_1^*) : \mathbf{Set}.}$$

Thus, there is a derivation

$$\frac{}{\mathbf{0}\ \mathsf{small}} \qquad \frac{}{\mathbf{1}\ \mathsf{small}} \qquad \frac{\mathcal{C}\ \mathsf{small}}{\mathcal{C}^{\mathsf{op}}\ \mathsf{small}}$$

$$\frac{\mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{small}}{\mathcal{C} \times \mathcal{D}\ \mathsf{small}} \qquad \frac{\mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{small}}{\mathcal{C} + \mathcal{D}\ \mathsf{small}}$$

$$\frac{}{\mathbf{Set}\ \mathsf{complete}} \qquad \frac{\mathcal{D}\ \mathsf{complete}}{[\mathcal{C}, \mathcal{D}]\ \mathsf{complete}}$$

Figure 1: Some rules for smallness and completeness.

$$\frac{\dfrac{\Gamma, X : \mathbb{C}^{\mathsf{op}}, Y : \mathbb{C} \vdash E : \mathcal{D}}{\Gamma^{\mathsf{op}}, Y : \mathbb{C}^{\mathsf{op}}, X : \mathbb{C} \vdash E^{*} : \mathcal{D}^{\mathsf{op}}}\ \mathsf{dua} + \mathsf{exc}}{\Gamma^{\mathsf{op}}, W : \mathcal{D} \vdash \mathcal{D}^{\mathsf{op}}(W, \int_{Y^{\mathbb{C}^{\mathsf{op}}}, X^{\mathbb{C}}} E^{*}) \cong \int_{Y^{\mathbb{C}^{\mathsf{op}}}, X^{\mathbb{C}}} \mathcal{D}^{\mathsf{op}}(W, E^{*}) : \mathbf{Set}.}\ \mathsf{end}$$

Using (opp) twice and transitivity we obtain the derived rule

$$\mathsf{end*}\ \frac{\Gamma, X : \mathbb{C}^{\mathsf{op}}, Y : \mathbb{C} \vdash E : \mathcal{D}}{\Gamma^{\mathsf{op}}, W : \mathcal{D} \vdash \mathcal{D}(\int^{X^{\mathbb{C}^{\mathsf{op}}}, Y^{\mathbb{C}}} E, W) \cong \int_{Y^{\mathbb{C}^{\mathsf{op}}}, X^{\mathbb{C}}} \mathcal{D}(E, W) : \mathbf{Set}.}$$

We have chosen to formalise dualisation in rules and then derive rules for coends. By starting out with sufficient extra rules for coends we could eliminate the dualisation rules from any derivation.

4 The Calculus

4.1 Rules for Typing

To avoid the set-theoretic paradoxes, categories are classified according to their size as small and locally small. The formalisation of this distinction demands a form of judgement for types: $\mathcal{C}$ $\mathsf{is\ small}$ where $\mathcal{C}$ is any type. Similarly, the rules involving end expressions make assumptions about completeness asking for judgements: $\mathcal{C}$ $\mathsf{is\ complete}$ and $\mathcal{C}$ $\mathsf{is\ cocomplete}$ where $\mathcal{C}$ is any type.

With a view to keeping the presentation of the rules compact, assumptions about types have been presented rather informally. In Fig. 1, however, the basic set of formal rules is shown. Rules for other constants than **Set** should be added. This point is not elaborated here; a natural extension to this work is to design a richer language for types.

In Fig. 2 we show the typing rules for expressions. These rules are interpreted as operations on functors, taking the functors denoted by the premises to that denoted by the conclusion. Such an interpretation yields a categorical semantics in the usual sense (see [1, 5, 8]).

The rule for assumption (ass) is interpreted as the identity functor which exists for any category. The interpretation of the rule for weakening (wea) takes the functor $\Gamma \xrightarrow{E} \mathcal{C}$ denoted by the premise of the rule to the composition $\Gamma \times \Gamma' \xrightarrow{\pi_\Gamma} \Gamma \xrightarrow{E} \mathcal{C}$ which denotes the conclusion. We can interpret the rules for exchange and contraction similarly.

The rule for pairs (pai) corresponds to the application of the product functor $- \times - : \mathbf{CAT} \times \mathbf{CAT} \to \mathbf{CAT}$ to the interpretation of E_1 and E_2. The projections (fst) and (snd) are just the composition with the projection arrows associated to the product. The rules for introduction and elimination of sums are interpreted through the functor $- + - : \mathbf{CAT} \times \mathbf{CAT} \to \mathbf{CAT}$ in a similar fashion. The rule (dua) expresses the application of the functor $(-)^*$ to $\Gamma \xrightarrow{E} \mathcal{C}$.

The abstraction rule (lam) is interpreted by "currying" the functor denoted by E, *i.e.*, a variable from the context is shifted into the expression. Symmetrically the evaluation functor justifies the rule (app). The rule for substitution (sub) corresponds to the composition of functors; recall substitution may involve dualisation – see § 3.5.

The rules (ten) and (uni) have a special status. Products in **Set** are introduced with a view to a more general model given by $\mathcal{V}$-enriched categories, where $\mathcal{V}$ is a monoidal closed category equipped with a tensor product $\otimes$. In the case of **Set**, the tensor product is the categorical product.

4.2 Rules for Natural Isomorphisms

The rules for natural isomorphisms are listed in Fig. 3. The structural rules for weakening, exchange and contraction are defined as usual and not shown. The rules describe how to build natural isomorphisms. For example, the premise of the rule (laml)

$$X_1 : \mathcal{C}_1, \cdots, X_n : \mathcal{C}_n, X : \mathcal{C} \vdash E_1 \cong E_2 : \mathcal{D}$$

$$\mathsf{ass}\ \frac{}{X\!:\!\mathcal{C} \vdash X\!:\!\mathcal{C}} \qquad\qquad \mathsf{dua}\ \frac{\Gamma \vdash E\!:\!\mathcal{D}}{\Gamma^{\mathsf{op}} \vdash E^{*}\!:\!\mathcal{D}^{\mathsf{op}}}$$

$$\mathsf{wea}\ \frac{\Gamma \vdash E\!:\!\mathcal{C}}{\Gamma, \Gamma' \vdash E\!:\!\mathcal{C}} \qquad\qquad \mathsf{exc}\ \frac{\Gamma \vdash E\!:\!\mathcal{C}}{\Pi(\Gamma) \vdash E\!:\!\mathcal{C}}\Pi \text{ permutation}$$

$$\mathsf{con}\ \frac{\Gamma, X\!:\!\mathcal{C}, Y\!:\!\mathcal{C} \vdash E\!:\!\mathcal{D}}{\Gamma, Z\!:\!\mathcal{C} \vdash E[Z/X, Z/Y]\!:\!\mathcal{D}}\ Z \text{ is fresh}$$

$$\mathsf{fst}\ \frac{\Gamma \vdash E\!:\!\mathcal{C} \times \mathcal{D}}{\Gamma \vdash \mathit{fst}(E)\!:\!\mathcal{C}} \qquad\qquad \mathsf{snd}\ \frac{\Gamma \vdash E\!:\!\mathcal{C} \times \mathcal{D}}{\Gamma \vdash \mathit{snd}(E)\!:\!\mathcal{D}}$$

$$\mathsf{pai}\ \frac{\Gamma \vdash E_1\!:\!\mathcal{C} \quad \Gamma' \vdash E_2\!:\!\mathcal{D}}{\Gamma, \Gamma' \vdash (E_1, E_2)\!:\!\mathcal{C} \times \mathcal{D}}$$

$$\mathsf{inl}\ \frac{\Gamma \vdash E\!:\!\mathcal{C}}{\Gamma \vdash \mathit{inl}(E)\!:\!\mathcal{C} + \mathcal{D}} \qquad\qquad \mathsf{inr}\ \frac{\Gamma \vdash E\!:\!\mathcal{D}}{\Gamma \vdash \mathit{inr}(E)\!:\!\mathcal{C} + \mathcal{D}}$$

$$\mathsf{cas}\ \frac{\Gamma \vdash E_1\!:\!\mathcal{C} + \mathcal{D} \quad \Gamma' \vdash E_2\!:\!\mathcal{E} \quad \Gamma'' \vdash E_3\!:\!\mathcal{E}}{\Gamma, \Gamma', \Gamma'' \vdash \mathit{case}_{\mathcal{C}+\mathcal{D}}(E_1, E_2, E_3)\!:\!\mathcal{E}}$$

$$\mathsf{lam}\ \frac{\Gamma, X\!:\!\mathcal{C} \vdash E\!:\!\mathcal{D} \quad \mathcal{C}\ \mathsf{small}}{\Gamma \vdash \lambda X^{\mathcal{C}}.E\!:\![\mathcal{C}, \mathcal{D}]} \qquad \mathsf{app}\ \frac{\Gamma \vdash F\!:\![\mathcal{C}, \mathcal{D}] \quad \Gamma' \vdash E\!:\!\mathcal{C} \quad \mathcal{C}\ \mathsf{small}}{\Gamma, \Gamma' \vdash F(E)\!:\!\mathcal{D}}$$

$$\mathsf{hom}\ \frac{\Gamma \vdash E_1\!:\!\mathcal{C} \quad \Gamma' \vdash E_2\!:\!\mathcal{C}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash \mathcal{C}(E_1, E_2)\!:\!\mathbf{Set}} \qquad \mathsf{sub}\ \frac{\Gamma, X\!:\!\mathcal{C} \vdash E_1\!:\!\mathcal{D} \quad \Gamma' \vdash E_2\!:\!\mathcal{C}}{\Gamma, \Gamma' \vdash E_1[E_2/X]\!:\!\mathcal{D}}$$

$$\mathsf{uni}\ \frac{}{\vdash 1\!:\!\mathbf{Set}} \qquad\qquad \mathsf{ten}\ \frac{\Gamma \vdash E_1\!:\!\mathbf{Set} \quad \Gamma' \vdash E_2\!:\!\mathbf{Set}}{\Gamma, \Gamma' \vdash E_1 \times E_2\!:\!\mathbf{Set}}$$

$$\mathsf{int}\ \frac{\Gamma, X\!:\!\mathcal{C}^{\mathsf{op}}, Y\!:\!\mathcal{C} \vdash E\!:\!\mathcal{D} \quad \mathcal{D}\ \mathsf{complete} \quad \mathcal{C}\ \mathsf{small}}{\Gamma \vdash \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} E\!:\!\mathcal{D}}$$

Figure 2: Typing rules.

$$\mathsf{ref}\ \frac{\Gamma \vdash E : \mathcal{C}}{\Gamma \vdash E \cong E : \mathcal{C}} \qquad \mathsf{sym}\ \frac{\Gamma \vdash E_1 \cong E_2 : \mathcal{C}}{\Gamma \vdash E_2 \cong E_1 : \mathcal{C}}$$

$$\mathsf{tra}\ \frac{\Gamma \vdash E_1 \cong E_2 : \mathcal{C} \quad \Gamma \vdash E_2 \cong E_3 : \mathcal{C}}{\Gamma \vdash E_1 \cong E_3 : \mathcal{C}} \qquad \mathsf{subl}\ \frac{\Gamma, X : \mathcal{C} \vdash E_1 \cong E_2 : \mathcal{D} \quad \Gamma' \vdash E : \mathcal{C}}{\Gamma, \Gamma' \vdash E_1[E/X] \cong E_2[E/X] : \mathcal{D}}$$

$$\mathsf{appl}\ \frac{\Gamma \vdash F \cong G : [\mathcal{C}, \mathcal{D}] \quad \Gamma' \vdash E : \mathcal{C} \quad \mathcal{C}\ \mathsf{small}}{\Gamma, \Gamma' \vdash F(E) \cong G(E) : \mathcal{D}} \qquad \mathsf{laml}\ \frac{\Gamma, X : \mathcal{C} \vdash E_1 \cong E_2 : \mathcal{D} \quad \mathcal{C}\ \mathsf{small}}{\Gamma \vdash \lambda X^{\mathcal{C}}.E_1 \cong \lambda X^{\mathcal{C}}.E_2 : [\mathcal{C}, \mathcal{D}]}$$

$$\mathsf{intl}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{C} \vdash E_1 \cong E_2 : \mathcal{D} \quad \mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{complete}}{\Gamma \vdash \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} E_1 \cong \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} E_2 : \mathcal{D}}$$

$$\mathsf{end}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{C} \vdash E : \mathcal{D} \quad \mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{complete}}{\Gamma, W : \mathcal{D}^{\mathsf{op}} \vdash \mathcal{D}\big(W, \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} E\big) \cong \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} \mathcal{D}(W, E) : \mathbf{Set}}$$

$$\mathsf{nat}\ \frac{\Gamma, X : \mathcal{C} \vdash E_1 : \mathcal{D} \quad \Gamma', Y : \mathcal{C} \vdash E_2 : \mathcal{D} \quad \mathcal{C}\ \mathsf{small}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash [\mathcal{C}, \mathcal{D}](\lambda X^{\mathcal{C}}.E_1, \lambda Y^{\mathcal{C}}.E_2) \cong \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} \mathcal{D}\big(E_1, E_2\big) : \mathbf{Set}}$$

$$\mathsf{fub}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{C}, W : \mathcal{D}^{\mathsf{op}}, Z : \mathcal{D} \vdash E : \mathcal{E} \quad \mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{small} \quad \mathcal{E}\ \mathsf{complete}}{\Gamma \vdash \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} \int_{W^{\mathcal{D}^{\mathsf{op}}}, Z^{\mathcal{D}}} E \cong \int_{W^{\mathcal{D}^{\mathsf{op}}}, Z^{\mathcal{D}}} \int_{X^{\mathcal{C}^{\mathsf{op}}}, Y^{\mathcal{C}}} E : \mathcal{E}}$$

$$\mathsf{dual}\ \frac{\Gamma \vdash E_1 \cong E_2 : \mathcal{D}}{\Gamma^{\mathsf{op}} \vdash E_1^* \cong E_2^* : \mathcal{D}^{\mathsf{op}}} \qquad \mathsf{opp} \frac{\Gamma \vdash E_1 : \mathcal{C} \quad \Gamma' \vdash E_2 : \mathcal{C}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash \mathcal{C}(E_1, E_2) \cong \mathcal{C}^{\mathsf{op}}(E_2^*, E_1^*) : \mathbf{Set}}$$

$$\mathsf{unil}\ \frac{\Gamma \vdash E : \mathbf{Set}}{\Gamma \vdash 1 \times E \cong E : \mathbf{Set}} \qquad \mathsf{com}\ \frac{\Gamma \vdash E_1 : \mathbf{Set} \quad \Gamma' \vdash E_2 : \mathbf{Set}}{\Gamma, \Gamma' \vdash E_1 \times E_2 \cong E_2 \times E_1 : \mathbf{Set}}$$

$$\mathsf{clo}\ \frac{\Gamma \vdash E_1 : \mathbf{Set} \quad \Gamma' \vdash E_2 : \mathbf{Set} \quad \Gamma'' \vdash E_3 : \mathbf{Set}}{\Gamma^{\mathsf{op}}, (\Gamma')^{\mathsf{op}}, \Gamma'' \vdash [E_1 \times E_2, E_3] \cong [E_1, [E_2, E_3]] : \mathbf{Set}}$$

$$\mathsf{yon}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}} \vdash E : \mathbf{Set} \quad \mathcal{C}\ \mathsf{small}}{\Gamma, Z : \mathcal{C}^{\mathsf{op}} \vdash E[Z/X] \cong [\mathcal{C}^{\mathsf{op}}, \mathbf{Set}](\lambda X^{\mathcal{C}^{\mathsf{op}}}.\mathcal{C}(X, Z), \lambda X^{\mathcal{C}}.E) : \mathbf{Set}}$$

$$\mathsf{rep}\ \frac{\Gamma, X : \mathcal{C}^{\mathsf{op}} \vdash \mathcal{C}(X, E_1) \cong \mathcal{C}(X, E_2) : \mathbf{Set}}{\Gamma \vdash E_1 \cong E_2 : \mathcal{C}} X \notin FV(E_1) \cup FV(E_2)$$

$$\mathsf{cur}\ \frac{\Gamma, X : \mathcal{C}, Y : \mathcal{D} \vdash E_1 : \mathcal{E} \quad \Gamma', X : \mathcal{C}, Y : \mathcal{D} \vdash E_2 : \mathcal{E} \quad \mathcal{C}\ \mathsf{small} \quad \mathcal{D}\ \mathsf{small}}{\Gamma^{\mathsf{op}}, \Gamma' \vdash [\mathcal{C} \times \mathcal{D}, \mathcal{E}]\big(\lambda X^{\mathcal{C}}, Y^{\mathcal{D}}.E_1, \lambda X^{\mathcal{C}}, Y^{\mathcal{D}}.E_2\big) \cong [\mathcal{C}, [\mathcal{D}, \mathcal{E}]]\big(\lambda X^{\mathcal{C}}.\lambda Y^{\mathcal{D}}.E_1, \lambda X^{\mathcal{C}}.\lambda Y^{\mathcal{D}}.E_2\big) : \mathbf{Set}}$$

Figure 3: Rules for natural isomorphisms.

means in more informal mathematical notation that there is an isomorphism

$$E_1(X_1, \cdots, X_n, X) \overset{\theta_{X_1,\cdots,X_n,X}}{\cong} E_2(X_1, \cdots, X_n, X)$$

natural in $X_1, \cdots, X_n, X$.

Thus we can abstract on the variable X to obtain an isomorphism

$$\lambda X.E_1(X_1, \cdots, X_n, X) \overset{\langle\theta_{X_1,\cdots,X_n,X}\rangle_X}{\cong} \lambda X.E_2(X_1, \cdots, X_n, X)$$

natural in $X_1, \cdots, X_n$. This justifies the conclusion of the rule (laml). The rules for end, duality and application are interpreted in a similar way, their soundness resting on the earlier discussion. Just as we could derive the dual rule (end*) in § 3.5, we can derive dual rules (intl*), (rep*) and (fub*).

We can derive dual rules (intl*), (rep*) and (fub*) in a similar manner to the derivation of (end*).

The rules (clo) and (com) arise from the closed structure of **Set**. Notice that we abbreviate hom-expressions over **Set** by using brackets, *e.g.* $[A, B]$ instead of $\mathbf{Set}(A, B)$. The "closed" structure of **CAT** gives the rule (cur) for currying. A rule for swapping arguments in functors is derivable from (cur), (nat) and (fub).

We can derive a rule for the covariant version of the Yoneda lemma. Take the small category $\mathbb{C}$ in the premise of the rule (yon) to be an opposite category $\mathbb{D}^{\mathsf{op}}$ to obtain:

$$\Gamma, Z : (\mathbb{D}^{\mathsf{op}})^{\mathsf{op}} \vdash E[Z/X] \cong [(\mathbb{D}^{\mathsf{op}})^{\mathsf{op}}, \mathbf{Set}](\lambda X^{(\mathbb{D}^{\mathsf{op}})^{\mathsf{op}}}.\mathbb{D}^{\mathsf{op}}(X, Z), \lambda X^{(\mathbb{D}^{\mathsf{op}})^{\mathsf{op}}}.E) : \mathbf{Set}$$

By applying (opp), (laml) and (homl) (and rewriting) we get:

$$\Gamma, Z : \mathbb{D} \vdash E[Z/X] \cong [\mathbb{D}, \mathbf{Set}](\lambda X^{\mathbb{D}}.\mathbb{D}(X, Z), \lambda X^{\mathbb{D}}.E) : \mathbf{Set}.$$

5 Examples

5.1 Continuity

The two driving notions in the previous sections have been functoriality and natural isomorphism. The example below shows that the calculus is rich enough to prove continuity.

Definition 5.1 (Preservation of Limits) *Let $F : \mathbb{I} \to \mathcal{C}$ be a functor with limiting cone $\langle k_I : \varprojlim_{\mathbb{I}} F \to F(I)\rangle_{I\in\mathbb{I}}$, a functor $G : \mathcal{C} \to \mathcal{D}$ preserves the limit of F iff $\langle G(k_I) : G(\varprojlim_{\mathbb{I}} F) \to G(F(I))\rangle_{I\in\mathbb{I}}$ is a limiting cone in $\mathcal{D}$. Preservation of colimits is defined dually.*

We say that G preserves $\mathbb{I}$-index limits if G preserves the limits of all functors in $[\mathbb{I}, \mathcal{C}]$. A functor which preserves all limits is called *continuous.*

One might expect to prove that a given functor preserves limits by showing an isomorphism of objects. This gives, however, a necessary but not sufficient condition.[4] The situation improves when the isomorphism is sufficiently natural however:

Theorem 5.2 *Assume that the categories $\mathcal{C}$ and $\mathcal{D}$ have limits for all functors with domain $\mathbb{I}$. The functor $G : \mathcal{C} \to \mathcal{D}$ preserves $\mathbb{I}$-indexed limits iff*

$$\varprojlim_{\mathbb{I}}(G \circ F) \cong G(\varprojlim_{\mathbb{I}} F),$$

natural in $F \in [\mathbb{I}, \mathcal{C}]$ (see [6] for a proof.).

This theorem supplies the key to proving continuity within the calculus. For example, that $\mathcal{C}(C, X)$ preserves limits in X is a direct consequence of the naturality formula.

Another result we can prove in the calculus is that right adjoints preserve limits. Given the judgements

$$X : \mathcal{C} \vdash E_1 : \mathcal{D} \quad \text{and} \quad Y : \mathcal{D} \vdash E_2 : \mathcal{C}$$

respectively denoting functors $E_1 : \mathcal{C} \to \mathcal{D}$ and $E_2 : \mathcal{D} \to \mathcal{C}$, an adjunction where, as functors, E_1 is the left adjoint and E_2 is the right adjoint consists of an isomorphism

$$X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{D} \vdash \mathcal{D}\big(E_1, Y\big) \cong \mathcal{C}(X, E_2) : \mathbf{Set}.$$

By Theorem 5.2 it is enough to prove

$$H : [\mathbb{I}, \mathcal{D}] \vdash E_2[\textstyle\int_{I^{\mathbb{I}}} H/Y] \cong \textstyle\int_{I^{\mathbb{I}}} E_2[H(I)/Y] : \mathcal{C}$$

where $\int_{I^{\mathbb{I}}} H(I)$ is the limit of H. The proof proceeds backwards, using (rep) to obtain the goal:

$$\mathcal{C}\big(X, E_2[\textstyle\int_{I^{\mathbb{I}}} H(I)/Y]\big) \cong \mathcal{C}\big(X, \textstyle\int_{I^{\mathbb{I}}} E_2[H(I)/Y]\big)$$

[4]There are examples of functors which preserve limiting objects but don't preserve the limiting cones, see [3, ex. 5.3.16(4)].

The derivation of this goal is sketched in the chain of isomorphisms:

$$\begin{aligned}
\mathcal{C}\big(X, E_2[\textstyle\int_{I^{\mathbb{I}}} H(I)/Y]\big) &\cong \mathcal{D}\big(E_1, \textstyle\int_{I^{\mathbb{I}}} H(I)\big) && \text{by the adjunction,}\\
&\cong \textstyle\int_{I^{\mathbb{I}}} \mathcal{D}\big(E_1, H(I)\big) && \text{by } (\mathsf{end}),\\
&\cong \textstyle\int_{I^{\mathbb{I}}} \mathcal{C}\big(X, E_2[H(I)/Y]\big) && \text{by the adjunction,}\\
&\cong \mathcal{C}\big(X, \textstyle\int_{I^{\mathbb{I}}} E_2[H(I)/Y]\big) && \text{by } (\mathsf{end}).
\end{aligned}$$

We conclude that the functor denoted by $Y : \mathcal{C} \vdash E_2 : \mathcal{D}$ preserves limits.

The dual result, *i.e.* left adjoints preserve colimits, follows from the rules opp and dual. From the definition of adjunction and by applying the rule opp twice together with transitivity we get

$$X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{D} \vdash \mathcal{C}^{\mathsf{op}}(E_2^*, X) \cong \mathcal{D}^{\mathsf{op}}\big(Y, E_1^*\big) : \mathbf{Set}$$

i.e. an adjunction where the functor denoted by E_1^* is the right adjoint. From the derivation above we can conclude

$$H : [\mathbb{I}^{\mathsf{op}}, \mathcal{C}^{\mathsf{op}}] \vdash E_1^*[\textstyle\int_{I^{\mathbb{I}^{\mathsf{op}}}} H(I)/X] \cong \textstyle\int_{I^{\mathbb{I}^{\mathsf{op}}}} E_1^*[H(I)/X] : \mathcal{D}^{\mathsf{op}}$$

and by applying the rule dual

$$H : [\mathbb{I}, \mathcal{C}] \vdash E_1[\textstyle\int^{I^{\mathbb{I}}} H(I)/X] \cong \textstyle\int^{I^{\mathbb{I}}} E_1[H(I)/X] : \mathcal{D}.$$

5.2 Pointwise Limits

Assuming a complete category $\mathcal{C}$ and given a judgement

$$\Gamma, X : \mathbb{I}, Y : \mathbb{J} \vdash E : \mathcal{C}$$

we can derive the expression

$$\Gamma \vdash \lambda Y^{\mathbb{J}}. \textstyle\int_{X^{\mathbb{I}}} E \cong \textstyle\int_{X^{\mathbb{I}}} \lambda Y^{\mathbb{J}}.E : [\mathbb{J}, \mathcal{C}]. \tag{5}$$

In other words, limits in functor categories are obtained pointwise.[5]

We first show

$$[\mathbb{J}, \mathcal{C}]\big(H, \lambda Y^{\mathbb{J}}. \textstyle\int_{X^{\mathbb{I}}} E\big) \cong [\mathbb{I}, [\mathbb{J}, \mathcal{C}]]\big(\lambda X^{\mathbb{I}}.H, \lambda X^{\mathbb{I}}.\lambda Y^{\mathbb{J}}.E\big)$$

[5]That is provided the pointwise limits exist, but we assume $\mathcal{C}$ complete.

where H:$[\mathbb{J},\mathcal{C}]^{\mathsf{op}}$. A derivation for this judgement is sketched in the following chain of isomorphisms:

$$\begin{aligned}
[\mathbb{J},\mathcal{C}]\big(H, \lambda Y^{\mathbb{J}}.\textstyle\int_{X^{\mathbb{I}}} E\big) &\cong \textstyle\int_{Y^{\mathbb{J}},Z^{\mathbb{J}^{\mathsf{op}}}} \mathcal{C}\big(H\,Z, \int_{X^{\mathbb{I}}} E\big) && \text{by } (\mathsf{nat}),\\
&\cong \textstyle\int_{Y^{\mathbb{J}},Z^{\mathbb{J}^{\mathsf{op}}}} \int_{X^{\mathbb{I}}} \mathcal{C}\big(H\,Z, E\big) && \text{by } (\mathsf{end}),\\
&\cong \textstyle\int_{X^{\mathbb{I}}} \int_{Y^{\mathbb{J}},Z^{\mathbb{J}^{\mathsf{op}}}} \mathcal{C}\big(H\,Z, E\big) && \text{by } (\mathsf{fub}),\\
&\cong \textstyle\int_{X^{\mathbb{I}}} [\mathbb{J},\mathcal{C}]\big(H, \lambda Y^{\mathbb{J}}.E\big) && \text{by } (\mathsf{nat}),\\
&\cong [\mathbb{I},[\mathbb{J},\mathcal{C}]]\big(\lambda X.H, \lambda X.\lambda Y.E\big) && \text{by } (\mathsf{nat}).
\end{aligned}$$

Now from the definition of limit (see § 3.4) we obtain

$$[\mathbb{J},\mathcal{C}]\big(H, \lambda Y.\textstyle\int_{X^{\mathbb{I}}} E\big) \cong [\mathbb{J},\mathcal{C}]\big(H, \int_{X^{\mathbb{I}}} \lambda Y.E\big)$$

and by (rep) we can conclude (5). From this and by using the rules (dua) and (dual) we can derive the dual result: colimits in functor categories are obtained pointwise.

5.3 The Density Formula

Functor categories of the form $[\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ have a number of important properties that give them a special rank in the calculus.[6] From the rules (yon) and (nat), a functor $X : [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]$ is expressible as the end formula:

$$X(W) \cong \textstyle\int_{Y^{\mathbb{C}^{\mathsf{op}}}} \big[\mathbb{C}(Y^-, W), X(Y^+)\big] \ . \tag{6}$$

There is also a coend formula for X, the so-called *density formula*:

$$X(W) \cong \textstyle\int^{Y^{\mathbb{C}}} \mathbb{C}(W, Y^+) \times X(Y^-) \ . \tag{7}$$

The chain of isomorphisms below sketches a derivation for this formula:

$$\begin{aligned}
&[\mathbb{C}^{\mathsf{op}},\mathbf{Set}]\big(\lambda W^{\mathbb{C}^{\mathsf{op}}}.\textstyle\int^{Y^{\mathbb{C}}} \mathbb{C}(W, Y^+) \times X(Y^-), Z\big)\\
&\quad\cong \textstyle\int_{W^{\mathbb{C}^{\mathsf{op}}}} [(\int^{Y^{\mathbb{C}}} \mathbb{C}(W^-, Y^+) \times X(Y^-), Z(W^+)] && \text{by } (\mathsf{nat}),\\
&\quad\cong \textstyle\int_{W^{\mathbb{C}^{\mathsf{op}}}} \int_{Y^{\mathbb{C}^{\mathsf{op}}}} [\mathbb{C}(W^-, Y^+) \times X(Y^-), Z(W^+)] && \text{by } (\mathsf{end*}) \text{ and } (\mathsf{intl}),\\
&\quad\cong \textstyle\int_{Y^{\mathbb{C}^{\mathsf{op}}}} \int_{W^{\mathbb{C}^{\mathsf{op}}}} [\mathbb{C}(W^-, Y^+) \times X(Y^-), Z(W^+)] && \text{by } (\mathsf{fub}),\\
&\quad\cong \textstyle\int_{Y^{\mathbb{C}^{\mathsf{op}}}} \int_{W^{\mathbb{C}^{\mathsf{op}}}} \big[X(Y^-), [\mathbb{C}(W^-, Y^+), Z(W^+)]\big] && \text{by } (\mathsf{com}) \text{ and } (\mathsf{clo})\\
&\quad\cong \textstyle\int_{Y^{\mathbb{C}^{\mathsf{op}}}} \big[X(Y^-), \int_{W^{\mathbb{C}^{\mathsf{op}}}} [\mathbb{C}(W^-, Y^+), Z(W^+)]\big] && \text{by } (\mathsf{end}) \text{ and } (\mathsf{intl}),\\
&\quad\cong \textstyle\int_{Y^{\mathbb{C}^{\mathsf{op}}}} [X(Y^-), Z(Y^+)] && \text{by (6) and } (\mathsf{intl}),\\
&\quad\cong [\mathbb{C}^{\mathsf{op}}, \mathbf{Set}]\big(X, Z\big) && \text{by } (\mathsf{nat}).
\end{aligned}$$

[6]Such categories are called presheaf categories.

The next step applies the rule (`rep*`) to yield the isomorphism (7).

6 Implementation in Isabelle

As part of ongoing research we are carrying out the implementation of the calculus in the theorem prover Isabelle. The calculus is implemented in several user-defined theories which extend the initial theory `Pure`. Basically, `Pure` gives the built-in higher-order logic [14] on top of which we define the object-logic for categories.

There are two meta-level types: `cat` and `exp` for the types and the expressions of the language respectively. The types constructors are defined as constants, for example:

```
Set    :: cat
One    :: cat
Op     :: cat => cat              ("_^op" [10] 10)
FunCat :: [cat, cat] => cat       ("[_,_]") .
```

The syntax for the expressions is defined similarly, for example:

```
Hom  :: [cat,exp,exp]=> exp
Lam  :: (exp=>exp) => exp         (binder "LAM" 10)
Into :: (exp=>exp) => exp         (binder "INTO" 10)
Int  :: (exp=>exp) => exp         (binder "INT" 10) .
```

where `Lam`, `Into` and `Int` give the definition for the binders λ and $\int$. There are two special constants to encode assertions for types and natural isomorphisms:

```
":"    :: [exp,cat] => o        (infixl 7)
NatIso :: [exp,exp,cat] => o    ("_~_:_" [5,5,5] 5) .
```

The constant "`:`" takes a term and a type as input and returns a truth value, similarly `NatIso` takes two terms and a type a returns a truth value.

The judgements of the calculus correspond to theorems in the meta-logic. For example, $X : \mathcal{C}^{\mathsf{op}}, Y : \mathcal{C} \vdash \mathcal{C}(X, Y) : \mathbf{Set}$ is written in the object-logic for categories as

```
!! X Y. [| X:C^op; Y:C |] ==> Hom(C,X,Y):Set .
```

Notice that the universal quantification prevents a possible instantiation of the meta-variables `X` and `Y` enforcing their treatment as variables in the object-logic level. The manipulation of the contexts in the rules is done through lifting. Thus, the definition of the rules is only concerned with the part of the context which changes. For example, the rule for typing end formulae is:

```
int " ( !!X Y. [| X:C^op ; Y:C; C Small |] ==> E(X,Y):D)
                                ==> INTO X. INT Y. E(X,Y):D".
```

This approach gives the structural rules for free; the rule (ass) corresponds to a proof by assumption, exchange to rotating the assumptions in a subgoal, and so on. We are currently developing an alternative implementation of the calculus in Isabelle/HOL [12] where the contexts are defined explicitly as sets of pairs. This approach should ease the implementation of duality.

7 Conclusions, Related Work and Future Directions

A calculus which formalises an expressive fragment of category theory has been presented. The rules of the calculus support a calculational approach to universality where the manipulation of ends and coends plays a central role. We have shown how to formalise judgements about functoriality and how to verify isomorphisms. In this setting, we have explored duality and studied applications to encode and prove results on continuous functors. An implementation of the calculus in the theorem prover Isabelle has been outlined briefly.

Previous work in the automation of category theory has followed a somewhat different tack. Takeyama [18] presents a computer checked language with the goal of supplying a categorical framework for type theory and includes an implementation in the theorem prover Lego [16]. This follows in spirit the work of Rydeheard and Burstall [17] on formalising universal constructions in ML. Beylin and Dybjer [4] formalise monoidal categories and present a proof of the Mac Lane's coherence theorem in Martin Löf's type theory [13].

The interpretation of the rules suggests a language for natural isomorphisms. Observe that in theorem provers like Isabelle where a goal may contain unknowns, a witness for a natural isomorphism may be given as output of the proof search. More generally, a further extension to the calculus is to introduce expressions for natural transformations.

A richer language for types may be considered by adding new constructors like recursion and lifting. For recursive definition of categories we need to extend the language for types to allow expressions with variables. The lifted type $\mathcal{C}_\perp$ would be interpreted as the category $\mathcal{C}$ with an initial object $\perp$ freely adjoined.

The utility of end and coend notation was demonstrated in the pioneering work of Kelly and others in enriched category theory [10]. In emphasising a calculus for categories, a goal has been to make working with functor categories routine. The formalisation here should lend itself to enriched category theory.

Acknowledgments: This work was inspired by Martin Hyland's Cambridge part III lecture course on category theory and its emphasis on end and coend manipulation.

References

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

[2] H. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2: Computational Structures. Oxford University Press, 1991.

[3] M. Barr and C. Wells. *Category Theory for Computing Science.* C.A.R. Hoare Series Editor. Prentice Hall, second edition, 1996.

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

[5] T. Braüner. *An Axiomatic Approach to Adequacy.* PhD thesis, BRICS PhD School, 1996.

[6] M. Cáccamo, J. M. E. Hyland, and G. Winskel. Lecture notes in category theory. Available from: `www.brics.dk/~mcaccamo`.

[7] G. L. Cattani. *Presheaf Models for Concurrency.* PhD thesis, BRICS PhD School, 1999.

[8] R. L. Crole. *Categories for Types.* Cambridge Mathematical Textbooks. Cambridge University Press, 1993.

[9] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation and open maps. In R. L. Constable, editor, *Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science*, Montreal, Canada, June 1993. IEEE Computer Society Press.

[10] G. M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1982.

[11] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer-Verlag, second edition, 1998.

[12] T. Nipkow and L. Paulson. Isabelle HOL: The tutorial. Documentation included in the official distribution of Isabelle, Feb. 2001.

[13] B. Nordström, K. Petterson, and J. Smith. *Programming in Martin-Löf's Type Theory. An Introduction.* Oxford University Press, 1990.

[14] L. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1988.

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

[16] R. Pollack. *The Theory of LEGO: A Proof Checker for the Extended Calculus of Constructions.* PhD thesis, The University of Edinburgh, 1994.

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

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