A CoEndoYoneda Lemma and its Relationship to a Pointfree Categorical Characterization of Purity

Luc Duponcheel

November 2026

Abstract

The classical CoYoneda lemma is a pointful statement about **SET**-valued *functors* $\mathrm{CYF}(Z) = \mathrm{Hom}_{\mathcal{C}}(Z, -) : \mathcal{C} \to \mathbf{SET}$; in the special case where the base category is $\mathcal{C} = \mathbf{SET}$ itself, $\mathrm{CYF}(Z) = \mathrm{Hom}_{\mathbf{SET}}(Z, -) : \mathbf{SET} \to \mathbf{SET}$ is already an *endofunctor*, and the classical CoYoneda lemma is a statement about endofunctors on **SET**. Motivated by pointfree functional programming—where programs are morphisms in a category $\mathcal{C}$ equipped with a *functional functor* $\mathrm{FF} : \mathbf{SET} \to \mathcal{C}$ lifting pure functions to morphisms—one can compose $\mathrm{CYF}(Z)$ with FF in *any* such category $\mathcal{C}$ to obtain an *endofunctor* $\mathrm{CYEF}(Z) = \mathrm{FF} \circ \mathrm{CYF}(Z) : \mathcal{C} \to \mathcal{C}$. We introduce *Basic Functional Categories* and *Functional Categories* relative to FF, equipped with a global-like endofunctor $G = \mathrm{CYEF}(\mathrm{FF}(T))$ (where $T = \{\mathrm{tt}^*\}$ is the terminal set), a global-element generator function $g_Z : Z \to \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z))$ in **SET** defined by $g_Z(z) = \mathrm{FF}(\mathrm{const}_z)$ for the constant function $\mathrm{const}_z : T \to Z$, a unit-like natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow G$ satisfying the parametric equational law $\eta_{\mathrm{FF}(Z)} = \mathrm{FF}(g_Z)$ (Equation (3)), and a multiplication-like natural transformation $\mu : G \circ G \Rightarrow G$ satisfying the left unit law $\eta_{G(Z)} \star \mu_Z = \mathrm{id}_{G(Z)}$ (Equation (5)). We then prove the *CoEndoYoneda Isomorphism Theorem* (Theorem 3.4): for every object $Z \in \mathrm{ob}(\mathcal{C})$ and endofunctor $\mathrm{EF} : \mathcal{C} \to \mathcal{C}$, there is a canonical bijection

$$\mathrm{NatTrans}\big(\mathrm{CYEF}(Z),\, G \circ \mathrm{EF}\big) \;\cong\; \mathrm{Hom}_{\mathcal{C}}\big(\mathrm{FF}(T),\, G(\mathrm{EF}(Z))\big),$$

which generalizes the classical CoYoneda lemma for endofunctors on **SET** (recovered when $\mathcal{C} = \mathbf{SET}$ and $\mathrm{FF} = \mathrm{Id}_{\mathbf{SET}}$, Proposition 3.6 and Subsection 3.4) to any Functional Category $(\mathcal{C}, \mathrm{FF})$. Evaluating the forward isomorphism at $\mathrm{EF} = \mathrm{Id}_{\mathcal{C}}$ and $Z = \mathrm{FF}(T)$ on the identity transformation id_G recovers the terminal unit component $\eta_{\mathrm{FF}(T)} = g_{\mathrm{FF}(T)}(\mathrm{id}_{\mathrm{FF}(T)}) = g_{\mathrm{FF}(T)}(g_T(\mathrm{tt}^*))$ (Proposition 3.5).

Next, we investigate which effectful categories of programs satisfy the Basic Functional Category axioms. For the Kleisli category $\mathbf{K}(M)$ of a monad $(M, \mathrm{mPure}, \mathrm{mApply})$ on **SET** equipped with the canonical identity-on-objects embedding $\mathrm{KFF} : \mathbf{SET} \to \mathbf{K}(M)$, we prove our second main result (Theorem 4.3): $(\mathbf{K}(M), \mathrm{KFF})$ *is a Basic Functional Category if and only if the monad M is idempotent* ($\mathrm{mMap}\ \mathrm{mPure} = \mathrm{mPure}$, equivalently $\mathrm{mApply} \circ (\mathrm{mPure} \circ \mathrm{mPure}) = \mathrm{mPure}$). Remarkably, even the specialized terminal unit law $\eta_{\mathrm{FF}(T)} = g_{\mathrm{FF}(T)}(g_T(\mathrm{tt}^*))$ (Proposition 2.5) suffices to force global idempotency of M, which in turn yields the full parametric law (3) definitionally. Thus, the Basic Functional Category law provides an elementary, pointfree, specification-level characterization of *computational purity*. We further prove the Borceux/Lawvere–Manes terminal subsingleton theorem ($M(T) \cong T$ for every pure monad, Theorem 4.4), provide complete equational classifications of seven standard computational effects (Option, List, Read R, Write W, State S, Cont R, and ErrorState E S, Theorem 5.1), prove a strict separation theorem (Theorem 5.2) with an explicit counterexample showing that Read R ($|R| \geq 2$) satisfies $M(T) \cong T$ while failing the Basic Functional Category condition, and discuss the scope, implications for commutative effects, and limitations of the framework (Subsection 5.2).

Provenance and Attribution Statement. The mathematical formulation of Basic Functional Categories, Functional Categories, the CoEndoYoneda Lemma, and the equational

proof proper of the CoEndoYoneda Isomorphism Theorem were conceived and proved by the author (Luc Duponcheel) [2]. By contrast, the Lean 4 mechanization and the discovery and proof of the exact equivalence between the Basic Functional Category structure on $\mathbf{K}(M)$ and monad idempotency/purity were established collaboratively using *Gemini* (Google DeepMind Antigravity) guided by mathematical prompts from the author. Both the Cubical Agda development [3] and the Palomar-verified Lean 4 formalization [4] are machine-checked with zero unproved axioms or postulates.

Keywords: CoYoneda lemma, pointfree category theory, functional categories, Kleisli categories, idempotent monads, computational purity, Cubical Agda, Lean 4.

MSC 2020: 18A40, 18C15, 18C20, 68N18, 68V20. **arXiv:** `cs.PL`, `math.CT`, `cs.LO`.

Contents

1 Introduction and Motivation **4**

1.1 From Pointful CoYoneda to Pointfree CoEndoYoneda 4

1.2 Theory Meets Practice: "Every Disadvantage Has Its Advantage" 5

1.3 Related Work on Categorical and Pointfree Characterizations of Purity 5

1.4 Provenance and Attribution of Results 6

2 Basic Functional Categories **6**

3 Functional Categories and the CoEndoYoneda Isomorphism **8**

3.1 Statement and Construction of the CoEndoYoneda Isomorphism 9

3.2 Connecting the CoEndoYoneda Isomorphism to η at $\mathrm{FF}(T)$ 11

3.3 Canonical Concrete Instance: **SET** as a Functional Category 11

3.4 Relationship to the Classical CoYoneda Lemma: Naturality, Universality, Generalization, and Specialization 12

4 Purity, Idempotent Monads, and Kleisli Categories **13**

4.1 Monad Formulations and Characterizations of Purity 13

4.2 Equivalence Between Kleisli Basic Functional Categories and Monad Idempotency 13

4.3 The Terminal Subsingleton Theorem 16

5 Classification of Concrete Computational Effects **17**

5.1 Strict Separation Between Basic Functional Category Purity and Terminal Subsingleton Triviality 19

5.2 Scope, Implications for Commutative Effects, and Limitations of the Framework 20

6 Formal Verification in Cubical Agda and Lean 4 **21**

1 Introduction and Motivation

1.1 From Pointful CoYoneda to Pointfree CoEndoYoneda

In classical category theory, CoYoneda theorems and proofs deal with **SET**-valued functors (co-presheaves)

$$\mathrm{CYF}(Z) : \mathcal{C} \longrightarrow \mathbf{SET}, \qquad Y \longmapsto \mathrm{Hom}_{\mathcal{C}}(Z, Y), \qquad (y2x : Y \to X) \longmapsto (z2y \mapsto z2y \star y2x),$$

where throughout this paper we write $f \star h : Z \to X$ for the diagrammatic composition of $f : Z \to Y$ followed by $h : Y \to X$ in $\mathcal{C}$. For any functor $F : \mathcal{C} \to \mathbf{SET}$, the classical CoYoneda lemma establishes a natural bijection

$$\mathrm{NatTrans}\big(\mathrm{CYF}(Z),\, F\big) \;\cong\; F(Z), \tag{1}$$

mediated by the universal element $\mathrm{id}_Z \in \mathrm{Hom}_{\mathcal{C}}(Z, Z) = \mathrm{CYF}(Z)(Z)$ via the forward map $\tau \mapsto \tau_Z(\mathrm{id}_Z)$ and the backward map $x \mapsto (Y \mapsto (z2y \mapsto F(z2y)(x)))$.

Notice an important special case of (1): *when the base category $\mathcal{C}$ is* **SET** *itself, the CoYoneda functor* $\mathrm{CYF}(Z) = \mathrm{Hom}_{\mathbf{SET}}(Z, -) : \mathbf{SET} \to \mathbf{SET}$ *and any target functor* $F = \mathrm{EF} : \mathbf{SET} \to \mathbf{SET}$ *are both* ***endofunctors*** *on* **SET**. Furthermore, in **SET**, the set of elements $\mathrm{EF}(Z)$ is canonically isomorphic to the hom-set of global elements $\mathrm{Hom}_{\mathbf{SET}}(T, \mathrm{EF}(Z)) = (T \to \mathrm{EF}(Z))$ out of the terminal singleton set $T = \{\mathrm{tt}^*\}$. Thus, on $\mathcal{C} = \mathbf{SET}$, the classical CoYoneda lemma (1) is already an *endofunctor* statement relating natural transformations between endofunctors on **SET** to morphisms out of T:

$$\mathrm{NatTrans}\big(\mathrm{CYF}(Z),\, \mathrm{EF}\big) \;\cong\; \mathrm{EF}(Z) \;\cong\; \mathrm{Hom}_{\mathbf{SET}}\big(T,\, \mathrm{EF}(Z)\big).$$

For a general category $\mathcal{C}$, however, classical CoYoneda theorems are inherently *pointful*: they study objects and functors by passing outside $\mathcal{C}$ into the category **SET** of hom-sets and set-theoretic elements. At the same time, they serve as the bridge from a pointful approach to a *pointfree* approach, characterizing objects by the morphisms out of them.

Motivated by *pointfree functional programming*—such as the author's Program Specification Based Programming (`PSBP`) architecture [5], where pure functions can be lifted to program morphisms—a natural foundational question arises:

> *What if the category $\mathcal{C}$ is also equipped with a "functional functor"* $\mathrm{FF} : \mathbf{SET} \to \mathcal{C}$ *mapping functions to morphisms?*

Whenever such a functor $\mathrm{FF} : \mathbf{SET} \to \mathcal{C}$ exists, every **SET**-valued CoYoneda functor $\mathrm{CYF}(Z) : \mathcal{C} \to \mathbf{SET}$ can be composed with FF to form an *endofunctor* on $\mathcal{C}$, just as in the $\mathcal{C} = \mathbf{SET}$ case:

$$\mathrm{CYEF}(Z) \;\coloneqq\; \mathrm{FF} \circ \mathrm{CYF}(Z) \;:\; \mathcal{C} \longrightarrow \mathcal{C}.$$

We can then ask: *Are there pointfree* ***CoEndoYoneda*** *theorems that generalize the $\mathcal{C} = \mathbf{SET}$ endofunctor CoYoneda lemma so that they can be formulated and proved entirely in terms of endofunctors on $\mathcal{C}$?*

In Sections 2–3, we show that this is indeed possible for categories equipped with $\mathrm{FF} : \mathbf{SET} \to \mathcal{C}$ together with two natural transformations (η and μ) and two simple equational laws (Equations (3) and (5)). In Subsection 3.4, we provide a detailed comparison of how the CoEndoYoneda isomorphism relates to the classical CoYoneda lemma in terms of naturality, universality, generalization, and specialization.

1.2 Theory Meets Practice: "Every Disadvantage Has Its Advantage"

Having established the CoEndoYoneda theorem, the natural practical question is: *For which categories of effectful programs can we apply this theorem?*

Originally, the author hoped that the CoEndoYoneda framework could be instantiated for at least a broad class of effectful (monadic) computations. However, testing the axioms on Kleisli categories $\mathbf{K}(M)$ revealed a surprise: the parametric unit law (3) (and even its terminal specialization (4) in Proposition 2.5) fails for impure effects such as Option, List, State S, and ErrorState E S. At first sight, this appeared to be bad news: the CoEndoYoneda bijection cannot be used for arbitrary effectful computations.

Yet, in the spirit of Willem van Hanegem's famous Dutch aphorism *"elk nadeel heb se voordeel"* (*"every disadvantage has its advantage"*, often misattributed to Johan Cruyff), this apparent limitation turned out to be a theoretical gift:

> *For a Kleisli category* $\mathbf{K}(M)$ *with its canonical identity-on-objects functional functor* $\mathrm{KFF} : \mathbf{SET} \to \mathbf{K}(M)$*, the Basic Functional Category structure is* ***equivalent*** *to the idempotency (purity) of the monad* M *(Theorem 4.3).*

While monads and their Kleisli extension / bind operator are inherently *pointful*, our Basic Functional Category axiom provides an elementary, *pointfree specification-level characterization of purity* directly on the category of programs.

1.3 Related Work on Categorical and Pointfree Characterizations of Purity

To position our contribution within the programming languages and category theory literature, we compare our Basic Functional Category characterization against established categorical frameworks for distinguishing pure and effectful computations:

(1) **Moggi's computational λ-calculus and strong monads [10]:** Moggi pioneered the modeling of computational effects via strong monads $(M, \eta, \mu, \mathrm{st})$ on a cartesian closed category $\mathcal{C}$. In Moggi's setting, pure programs are identified extrinsically as those Kleisli morphisms in $\mathbf{K}(M)$ that factor through the canonical inclusion functor $\mathrm{KFF} : \mathcal{C} \to \mathbf{K}(M)$, whereas our Theorem 4.3 characterizes when the entire Kleisli category $\mathbf{K}(M)$ collapses to pure computation via the existence of the CoYoneda-induced natural transformation $\eta : \mathrm{Id}_{\mathbf{K}(M)} \Rightarrow G$.

(2) **Premonoidal categories, Freyd categories, and central morphisms [15, 16, 9]:** Power, Robinson, and Thielecke (and Levy in Fine-Grain Call-by-Value) axiomatize effectful categories directly without mentioning a monad via *premonoidal categories* and *Freyd categories* (an identity-on-objects symmetric premonoidal functor $J : \mathcal{V} \to \mathcal{K}$ from a cartesian category of values $\mathcal{V}$ to a premonoidal category of computations $\mathcal{K}$). In a premonoidal category, a morphism $f : X \to Y$ is characterized pointfreely as *central* (effect-commuting) if $(f \ltimes g) = (f \rtimes g)$ for all morphisms g. While centrality captures commutativity of effects (and hence purity relative to non-commuting effects), it requires symmetric premonoidal tensor structure on $\mathcal{K}$ and still admits commutative effects (such as Read R or commutative Write W) in the center. By contrast, our Basic Functional Category condition requires no monoidal or premonoidal tensor structure on $\mathcal{C}$, and eliminates even commutative effects such as Read R (for $|R| \geq 2$) and commutative Write W (for $|W| \geq 2$).

(3) **Führmann's direct models of the computational λ-calculus and thunkability [6]:** Führmann gave a pointfree categorical semantics of call-by-value programming by equipping a category $\mathcal{K}$ with a *thunking endofunctor* $T : \mathcal{K} \to \mathcal{K}$ carrying a comonad-like

structure, characterizing pure ("thunkable") morphisms $f : X \to Y$ pointfreely by commutation with thunking ($f \star \vartheta_Y = \vartheta_X \star T(f)$, copyability, and discardability). Our global-like endofunctor $G = \mathrm{FF} \circ \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), -)$ plays a role analogous to an intrinsic thunking functor $Y \mapsto (T \to Y)$ constructed canonically from FF and $\mathrm{CYF}(\mathrm{FF}(T))$. However, whereas Führmann axiomatizes thunkable morphisms inside a general effectful category using comonadic structural laws, our Definition 2.4 requires only a single natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow G$ and the single equation $\eta_{\mathrm{FF}(Z)} = \mathrm{FF}(g)$ (or $\eta_{\mathrm{FF}(T)} = g(g(\mathrm{tt}^*))$).

(4) **Lawvere theories and algebraic effects [8, 13, 14]:** In the algebraic theory of effects developed by Plotkin and Power, finitary monads on **SET** correspond to Lawvere theories $\mathcal{L}$, and effectful operations are modeled by generic effects or algebraic operations. Within this framework, idempotent monads on **SET** correspond to *subterminal* (or propositional) Lawvere theories in which every n-ary operation reduces to a projection or constant, and on general categories idempotent monads correspond to reflective subcategories and well-pointed endofunctors ($M\eta = \eta M$) [7, 1, 11, 12]. Our Theorem 4.3 bridges this classical monadic characterization ($M\eta = \eta M$) with the pointfree CoEndoYoneda structure on $\mathbf{K}(M)$.

1.4 Provenance and Attribution of Results

In accordance with transparent human–AI scientific attribution standards (aiXiv and the Palomar Registry), we state the exact provenance of each contribution in this work:

(1) **Human-authored core theory and proof proper:** The categorical definitions of Basic Functional Categories and Functional Categories, the statement of the CoEndoYoneda Isomorphism Theorem (Theorem 3.4), and its complete pointfree equational proof proper were conceived, designed, and proved by the author, Luc Duponcheel [2].

(2) **AI-assisted purity equivalence and formal mechanizations:** The Lean 4 formalization and the discovery and proof of the exact equivalence between the Basic Functional Category structure on $\mathbf{K}(M)$ and monad idempotency/purity (Theorem 4.3), the Borceux/Lawvere–Manes terminal subsingleton diagnostic (Theorem 4.4), the formal classification of the seven computational effects and strict separation theorem (Section 5), and the synchronized Cubical Agda and Lean 4 implementations were developed using **Gemini** (Google DeepMind Antigravity) based upon targeted mathematical prompts and iterative guidance from the author.

2 Basic Functional Categories

Conventions and Notation. Throughout this paper, let $\mathcal{C}$ be a locally small category at universe level ℓ with object collection $\mathrm{ob}(\mathcal{C})$, and for objects $Z, Y \in \mathrm{ob}(\mathcal{C})$, let $\mathrm{Hom}_{\mathcal{C}}(Z, Y) \in \mathrm{ob}(\mathbf{SET}_\ell)$ denote the hom-set of morphisms in $\mathcal{C}$ from Z to Y. Because $\mathcal{C}$ is locally small at level ℓ, every hom-set $\mathrm{Hom}_{\mathcal{C}}(Z, Y)$ is an object of the category of sets $\mathbf{SET}_\ell$, and a morphism f from Z to Y in $\mathcal{C}$ is an element of that hom-set: we write $f : Z \to Y$ in $\mathcal{C}$ and $f \in \mathrm{Hom}_{\mathcal{C}}(Z, Y)$ synonymously.[1] We write $\mathrm{id}_Z : Z \to Z$ for the identity morphism on Z, and $f \star h : Z \to X$ for the *diagrammatic composition* of $f : Z \to Y$ followed by $h : Y \to X$ in an abstract category $\mathcal{C}$ (for Kleisli categories $\mathbf{K}(M)$ in Section 4, we write $\star_{\mathbf{K}}$ for Kleisli composition to distinguish it from

[1] In dependent type-theoretic proof assistants such as Cubical Agda and Lean 4, the hom-set $\mathrm{Hom}_{\mathcal{C}}(Z, Y)$ is represented as a type in $\mathrm{Type}\ \ell$, so set-theoretic membership $f \in \mathrm{Hom}_{\mathcal{C}}(Z, Y)$ is expressed by the typing judgment $f : \mathrm{Hom}_{\mathcal{C}}(Z, Y)$. In this paper we follow standard categorical notation $f : Z \to Y$ in $\mathcal{C}$ (or $f \in \mathrm{Hom}_{\mathcal{C}}(Z, Y)$) to keep the distinction between an individual morphism f and the hom-set $\mathrm{Hom}_{\mathcal{C}}(Z, Y) \in \mathrm{ob}(\mathbf{SET}_\ell)$ completely explicit.

ordinary function composition $\circ$ in $\mathbf{SET}_\ell$). For a functor $F : \mathcal{C} \to \mathcal{D}$, we write $F(Z) \in \mathrm{ob}(\mathcal{D})$ for its action on an object $Z \in \mathrm{ob}(\mathcal{C})$ and $F(f) : F(Z) \to F(Y)$ for its action on a morphism $f : Z \to Y$. Given functors $F, H : \mathcal{C} \to \mathcal{D}$, a natural transformation $\tau : F \Rightarrow H$ (an element of $\mathrm{NatTrans}(F, H)$) assigns to each object $Y \in \mathrm{ob}(\mathcal{C})$ a component morphism $\tau_Y : F(Y) \to H(Y)$ in $\mathcal{D}$ such that $F(f) \star \tau_X = \tau_Y \star H(f)$ for every morphism $f : Y \to X$ in $\mathcal{C}$. Finally, $\mathbf{SET}_\ell$ (abbreviated **SET** when the universe level ℓ is clear from context) denotes the category of sets at universe level ℓ (hSets in Cubical Agda, Type u in Lean 4), in which objects are sets and morphisms $f : Z \to Y$ are functions.

Definition 2.1. Given a locally small category $\mathcal{C}$ and a *functional functor* $\mathrm{FF} : \mathbf{SET}_\ell \to \mathcal{C}$, we define:

(i) The *CoYoneda embedding* $\mathrm{CYF} : \mathrm{ob}(\mathcal{C}) \to \mathrm{Functor}(\mathcal{C}, \mathbf{SET}_\ell)$: for each $Z \in \mathrm{ob}(\mathcal{C})$,

$$\begin{aligned} \mathrm{CYF}(Z)(Y) &\coloneqq \mathrm{Hom}_{\mathcal{C}}(Z, Y) \ \in \ \mathrm{ob}(\mathbf{SET}_\ell), \\ \mathrm{CYF}(Z)(y2x) &\coloneqq (z2y \mapsto z2y \star y2x) \ : \ \mathrm{Hom}_{\mathcal{C}}(Z, Y) \to \mathrm{Hom}_{\mathcal{C}}(Z, X). \end{aligned}$$

(ii) The *CoYoneda endofunctor* $\mathrm{CYEF} : \mathrm{ob}(\mathcal{C}) \to \mathrm{Functor}(\mathcal{C}, \mathcal{C})$:

$$\mathrm{CYEF}(Z) \ \coloneqq \ \mathrm{FF} \circ \mathrm{CYF}(Z) \ : \ \mathcal{C} \longrightarrow \mathcal{C}.$$

(iii) The *terminal set* $T \coloneqq \{\mathrm{tt}^*\} \in \mathrm{ob}(\mathbf{SET}_\ell)$, a singleton set with unique element $\mathrm{tt}^* \in T$.

(iv) The *global-like endofunctor* $G : \mathcal{C} \to \mathcal{C}$:

$$G \ \coloneqq \ \mathrm{CYEF}\big(\mathrm{FF}(T)\big) \ = \ \mathrm{FF} \circ \mathrm{CYF}\big(\mathrm{FF}(T)\big),$$

so that $G(Y) = \mathrm{FF}\big(\mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), Y)\big)$ for $Y \in \mathrm{ob}(\mathcal{C})$ and $G(y2x) = \mathrm{FF}(f \mapsto f \star y2x) : G(Y) \to G(X)$ for $y2x : Y \to X$.

(v) The *global-element generator* (a family of functions in $\mathbf{SET}_\ell$ indexed by sets $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$): for each set $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ and each element $z \in Z$, let $\mathrm{const}_z : T \to Z$ denote the constant function $\mathrm{const}_z(t) \coloneqq z$ in $\mathbf{SET}_\ell$. Applying the functor $\mathrm{FF} : \mathbf{SET}_\ell \to \mathcal{C}$ to the morphism $\mathrm{const}_z : T \to Z$ in $\mathbf{SET}_\ell$ yields a morphism $\mathrm{FF}(\mathrm{const}_z) : \mathrm{FF}(T) \to \mathrm{FF}(Z)$ in $\mathcal{C}$, i.e., an element of the hom-set $\mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z)) \in \mathrm{ob}(\mathbf{SET}_\ell)$. We therefore define the function $g_Z : Z \to \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z))$ in $\mathbf{SET}_\ell$ (often written simply g when the domain set Z is clear from context) by:

$$\begin{aligned} g_Z \ &: \ Z \longrightarrow \mathrm{Hom}_{\mathcal{C}}\big(\mathrm{FF}(T),\, \mathrm{FF}(Z)\big) \quad \text{in } \mathbf{SET}_\ell, \\ g_Z(z) \ &\coloneqq \ \mathrm{FF}(\mathrm{const}_z) \ \in \ \mathrm{Hom}_{\mathcal{C}}\big(\mathrm{FF}(T),\, \mathrm{FF}(Z)\big). \end{aligned} \tag{2}$$

Remark 2.2 (Two-level typing of g_Z and $\mathrm{FF}(g_Z)$). Because the construction of g_Z and $\mathrm{FF}(g_Z)$ crosses back and forth between $\mathbf{SET}_\ell$ and $\mathcal{C}$ via the functor $\mathrm{FF} : \mathbf{SET}_\ell \to \mathcal{C}$, it is helpful to trace the exact types at each step:

(1) For each element $z \in Z$, $\mathrm{const}_z = (_ \mapsto z) : T \to Z$ is a function (a morphism in $\mathbf{SET}_\ell$).

(2) Applying the functor FF to const_z produces a morphism $g_Z(z) = \mathrm{FF}(\mathrm{const}_z) : \mathrm{FF}(T) \to \mathrm{FF}(Z)$ in $\mathcal{C}$. Viewed as an element of the hom-set, $g_Z(z) \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z)) = \mathrm{CYF}(\mathrm{FF}(T))(\mathrm{FF}(Z))$.

(3) Consequently, the assignment $z \mapsto g_Z(z)$ is a well-defined *function in* $\mathbf{SET}_\ell$ from the set $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ to the hom-set $\mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z)) \in \mathrm{ob}(\mathbf{SET}_\ell)$:

$$g_Z \ \in \ \mathrm{Hom}_{\mathbf{SET}_\ell}\Big(Z,\, \mathrm{Hom}_{\mathcal{C}}\big(\mathrm{FF}(T), \mathrm{FF}(Z)\big)\Big) \ = \ \mathrm{Hom}_{\mathbf{SET}_\ell}\Big(Z,\, \mathrm{CYF}\big(\mathrm{FF}(T)\big)\big(\mathrm{FF}(Z)\big)\Big).$$

(4) Finally, because g_Z is a morphism in $\mathbf{SET}_\ell$, we can apply the functor $\mathrm{FF} : \mathbf{SET}_\ell \to \mathcal{C}$ a *second* time—now to the function g_Z itself—to obtain a morphism in $\mathcal{C}$:

$$\mathrm{FF}(g_Z) \ : \ \mathrm{FF}(Z) \longrightarrow \mathrm{FF}\Big(\mathrm{Hom}_{\mathcal{C}}\big(\mathrm{FF}(T), \mathrm{FF}(Z)\big)\Big) \ = \ G\big(\mathrm{FF}(Z)\big) \quad \text{in } \mathcal{C}.$$

Notice that $\mathrm{FF}(g_Z)$ has the exact same domain $\mathrm{FF}(Z)$ and codomain $G(\mathrm{FF}(Z))$ in $\mathcal{C}$ as the component $\eta_{\mathrm{FF}(Z)} : \mathrm{FF}(Z) \to G(\mathrm{FF}(Z))$ of any natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow G$. Equation (3) below simply asserts that these two morphisms in $\mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(Z), G(\mathrm{FF}(Z)))$ are equal.

Lemma 2.3. *For every element $t \in T$, $g_T(t) = \mathrm{id}_{\mathrm{FF}(T)}$. In particular, $g_T(\mathrm{tt}^*) = \mathrm{id}_{\mathrm{FF}(T)}$.*

Proof. Since $T = \{\mathrm{tt}^*\}$ has a unique inhabitant, any element $t \in T$ satisfies $t = \mathrm{tt}^*$, and the constant function $\mathrm{const}_t = (_ \mapsto t) : T \to T$ is extensionally equal to the identity function $(z \mapsto z) = \mathrm{id}_T : T \to T$. By functoriality of FF (preservation of identities),

$$g_T(t) \;=\; \mathrm{FF}(\mathrm{const}_t) \;=\; \mathrm{FF}(\mathrm{id}_T) \;=\; \mathrm{id}_{\mathrm{FF}(T)}.$$

Instantiating at $t = \mathrm{tt}^*$ gives $g_T(\mathrm{tt}^*) = \mathrm{id}_{\mathrm{FF}(T)}$. □

Definition 2.4. A *Basic Functional Category* over $(\mathcal{C}, \mathrm{FF})$ consists of:

(i) A "monad unit-like" natural transformation $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow G$ between the identity endofunctor $\mathrm{Id}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$ and the global-like endofunctor $G : \mathcal{C} \to \mathcal{C}$, with component morphisms

$$\eta_Y \;:\; Y \longrightarrow G(Y) \qquad (\text{i.e., } \eta_Y \in \mathrm{Hom}_{\mathcal{C}}(Y, G(Y)))$$

for each $Y \in \mathrm{ob}(\mathcal{C})$, satisfying the naturality condition $f \star \eta_X = \eta_Y \star G(f)$ for every morphism $f : Y \to X$ in $\mathcal{C}$;

(ii) The general parametric equational law: for every set $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$,

$$\eta_{\mathrm{FF}(Z)} \;=\; \mathrm{FF}(g_Z) \quad : \quad \mathrm{FF}(Z) \longrightarrow G\big(\mathrm{FF}(Z)\big) \quad \text{in } \mathcal{C}, \tag{3}$$

where $g_Z : Z \to \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(Z))$ is the global-element generator function in $\mathbf{SET}_\ell$ from Definition 2.1(v) and Remark 2.2.

Proposition 2.5. *In any Basic Functional Category $(\mathcal{C}, \mathrm{FF})$, let $H_T \coloneqq \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(T)) \in \mathrm{ob}(\mathbf{SET}_\ell)$. Then the component of η at $\mathrm{FF}(T)$ satisfies the terminal unit law:*

$$\eta_{\mathrm{FF}(T)} \;=\; g_{H_T}\big(g_T(\mathrm{tt}^*)\big) \;=\; g_{H_T}\big(\mathrm{id}_{\mathrm{FF}(T)}\big) \quad : \quad \mathrm{FF}(T) \longrightarrow G\big(\mathrm{FF}(T)\big), \tag{4}$$

or in subscript-free shorthand, $\eta_{\mathrm{FF}(T)} = g(g(\mathrm{tt}^*)) = g(\mathrm{id}_{\mathrm{FF}(T)})$.

Proof. Instantiating the general parametric law (3) at $Z = T$ gives $\eta_{\mathrm{FF}(T)} = \mathrm{FF}(g_T)$, where $g_T : T \to H_T$ with $H_T = \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), \mathrm{FF}(T))$. Because $T = \{\mathrm{tt}^*\}$ is the terminal set, every $t \in T$ equals tt^*, so $g_T : T \to H_T$ is equal to the constant function $\mathrm{const}_{g_T(\mathrm{tt}^*)} = (_ \mapsto g_T(\mathrm{tt}^*)) : T \to H_T$. By Definition 2.1(v) applied to the set $H_T \in \mathrm{ob}(\mathbf{SET}_\ell)$ and the element $g_T(\mathrm{tt}^*) \in H_T$, we have

$$\eta_{\mathrm{FF}(T)} \;=\; \mathrm{FF}(g_T) \;=\; \mathrm{FF}\big(\mathrm{const}_{g_T(\mathrm{tt}^*)}\big) \;=\; g_{H_T}\big(g_T(\mathrm{tt}^*)\big).$$

Finally, $g_T(\mathrm{tt}^*) = \mathrm{id}_{\mathrm{FF}(T)}$ by Lemma 2.3, so $g_{H_T}(g_T(\mathrm{tt}^*)) = g_{H_T}(\mathrm{id}_{\mathrm{FF}(T)})$. □

3 Functional Categories and the CoEndoYoneda Isomorphism

Definition 3.1. A *Functional Category* over $(\mathcal{C}, \mathrm{FF})$ is a Basic Functional Category over $(\mathcal{C}, \mathrm{FF})$ equipped with:

(i) A "monad multiplication-like" natural transformation $\mu : G \circ G \Rightarrow G$, with component morphisms

$$\mu_Y \ : \ G(G(Y)) \longrightarrow G(Y) \qquad (\text{i.e., } \mu_Y \in \mathrm{Hom}_{\mathcal{C}}(G(G(Y)), G(Y)))$$

for each $Y \in \mathrm{ob}(\mathcal{C})$, satisfying the naturality condition $G(G(f)) \star \mu_X = \mu_Y \star G(f)$ for every morphism $f : Y \to X$ in $\mathcal{C}$;

(ii) The monadic left identity law: for every object $Z \in \mathrm{ob}(\mathcal{C})$,

$$\eta_{G(Z)} \star \mu_Z \ = \ \mathrm{id}_{G(Z)} \quad : \quad G(Z) \longrightarrow G(Z). \tag{5}$$

Remark 3.2. Notice that neither the right unit law $G(\eta_Z) \star \mu_Z = \mathrm{id}_{G(Z)}$ nor the associativity law $\mu_{G(Z)} \star \mu_Z = G(\mu_Z) \star \mu_Z$ is required to establish the CoEndoYoneda isomorphism below: only the left identity law (5) is needed.

3.1 Statement and Construction of the CoEndoYoneda Isomorphism

Fix a Functional Category $(\mathcal{C}, \mathrm{FF})$, an object $Z \in \mathrm{ob}(\mathcal{C})$, and an endofunctor $\mathrm{EF} : \mathcal{C} \to \mathcal{C}$. Set:

$$G \circ \mathrm{EF} \ \coloneqq \ G \circ \mathrm{EF} : \mathcal{C} \to \mathcal{C}, \qquad \mathrm{CYF} \circ G \circ \mathrm{EF} \ \coloneqq \ \mathrm{CYF}(\mathrm{FF}(T)) \circ G \circ \mathrm{EF} : \mathcal{C} \to \mathbf{SET}_\ell.$$

Observe that the set $(\mathrm{CYF} \circ G \circ \mathrm{EF})(Z)$ is the hom-set of global-like elements of $(G \circ \mathrm{EF})(Z)$:

$$(\mathrm{CYF} \circ G \circ \mathrm{EF})(Z) \ = \ \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T),\, G(\mathrm{EF}(Z))).$$

In the classical CoYoneda lemma (1), the forward isomorphism evaluates a natural transformation $\tau : \mathrm{CYF}(Z) \Rightarrow F$ on the identity morphism $\mathrm{id}_Z \in \mathrm{Hom}_{\mathcal{C}}(Z, Z) = \mathrm{CYF}(Z)(Z)$. In our pointfree setting, we lift the element $\mathrm{id}_Z \in \mathrm{Hom}_{\mathcal{C}}(Z, Z)$ to a global-element morphism $\iota_Z : \mathrm{FF}(T) \to \mathrm{CYEF}(Z)(Z)$ in $\mathcal{C}$ via the global-element generator (2) on the set $\mathrm{Hom}_{\mathcal{C}}(Z, Z) \in \mathrm{ob}(\mathbf{SET}_\ell)$:

$$\iota_Z \ \coloneqq \ g_{\mathrm{Hom}_{\mathcal{C}}(Z,Z)}(\mathrm{id}_Z) \ = \ \mathrm{FF}(\mathrm{const}_{\mathrm{id}_Z}) \ : \ \mathrm{FF}(T) \longrightarrow \mathrm{CYEF}(Z)(Z), \tag{6}$$

where $\mathrm{const}_{\mathrm{id}_Z} = (_ \mapsto \mathrm{id}_Z) : T \to \mathrm{Hom}_{\mathcal{C}}(Z, Z)$ and $\mathrm{CYEF}(Z)(Z) = \mathrm{FF}(\mathrm{Hom}_{\mathcal{C}}(Z, Z))$.

Definition 3.3. We define two maps between $\mathrm{NatTrans}(\mathrm{CYEF}(Z), G \circ \mathrm{EF})$ and $(\mathrm{CYF} \circ G \circ \mathrm{EF})(Z)$:

(i) **Forward extraction map** to $= (\tau_Z \mapsto f_Z)$: given a natural transformation $\tau_Z : \mathrm{CYEF}(Z) \Rightarrow G \circ \mathrm{EF}$, define the morphism

$$\tau_Z \to f_Z \ \coloneqq \ \iota_Z \star (\tau_Z)_Z \ : \ \mathrm{FF}(T) \longrightarrow G(\mathrm{EF}(Z)) \quad \text{in } \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T),\, G(\mathrm{EF}(Z))). \tag{7}$$

(ii) **Backward reconstruction map** from $= (f_Z \mapsto \tau_Z)$: given $f_Z \in \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), G(\mathrm{EF}(Z)))$, define for each object $Y \in \mathrm{ob}(\mathcal{C})$ the auxiliary function in $\mathbf{SET}_\ell$

$$\alpha_{f_Z,Y} : \mathrm{Hom}_{\mathcal{C}}(Z, Y) \longrightarrow \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T),\, G(\mathrm{EF}(Y))), \qquad z2y \longmapsto f_Z \star (G \circ \mathrm{EF})(z2y),$$

and set the component morphism $(f_Z \to \tau_Z)_Y : \mathrm{CYEF}(Z)(Y) \to G(\mathrm{EF}(Y))$ in $\mathcal{C}$ to be

$$(f_Z \to \tau_Z)_Y \ \coloneqq \ \mathrm{FF}(\alpha_{f_Z,Y}) \star \mu_{\mathrm{EF}(Y)} \ : \ \mathrm{CYEF}(Z)(Y) \longrightarrow G(\mathrm{EF}(Y)). \tag{8}$$

Theorem 3.4. *In any Functional Category $(\mathcal{C}, \mathrm{FF})$, for every object $Z \in \mathrm{ob}(\mathcal{C})$ and every endofunctor $\mathrm{EF} : \mathcal{C} \to \mathcal{C}$:*

(i) For every $f_Z \in (\mathrm{CYF} \circ G \circ \mathrm{EF})(Z)$, the family $(f_Z \to \tau_Z)_Y$ defined in (8) is natural in Y, defining a natural transformation $(f_Z \to \tau_Z) : \mathrm{CYEF}(Z) \Rightarrow G \circ \mathrm{EF}$.

(ii) ***Section:*** *For every* $f_Z \in (\mathrm{CYF} \circ G \circ \mathrm{EF})(Z)$,

$$(f_Z \to \tau_Z) \to f_Z \;=\; f_Z.$$

(iii) ***Retraction:*** *For every* $\tau_Z \in \mathrm{NatTrans}(\mathrm{CYEF}(Z), G \circ \mathrm{EF})$,

$$(\tau_Z \to f_Z) \to \tau_Z \;=\; \tau_Z.$$

Consequently, $(\tau_Z \to f_Z)$ *and* $(f_Z \to \tau_Z)$ *form a bijection (isomorphism of types)*

$$\mathrm{NatTrans}(\mathrm{CYEF}(Z),\, G \circ \mathrm{EF}) \;\cong\; \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T),\, G(\mathrm{EF}(Z))).$$

Proof. We follow the equational chains formalized in [3] and mechanized in [4].

Part (i): Naturality of $f_Z \to \tau_Z$. First, for any morphisms $y2x : Y \to X$ and $z2y : Z \to Y$ in $\mathcal{C}$, functoriality of $G \circ \mathrm{EF}$ and associativity of composition in $\mathcal{C}$ give:

$$\begin{aligned} f_Z \star (G \circ \mathrm{EF})(z2y \star y2x) &= f_Z \star \big((G \circ \mathrm{EF})(z2y) \star (G \circ \mathrm{EF})(y2x)\big) \\ &= \big(f_Z \star (G \circ \mathrm{EF})(z2y)\big) \star (G \circ \mathrm{EF})(y2x). \end{aligned}$$

Now let $y2x : Y \to X$ in $\mathcal{C}$. Using $\mathrm{CYEF}(Z)(y2x) = \mathrm{FF}(z2y \mapsto z2y \star y2x)$, functoriality of FF on composite functions, the definition $(G \circ G)(\mathrm{EF}(y2x)) = \mathrm{FF}(h \mapsto h \star (G \circ \mathrm{EF})(y2x))$, and naturality of μ at $\mathrm{EF}(y2x)$:

$$\begin{aligned} &\mathrm{CYEF}(Z)(y2x) \star (f_Z \to \tau_Z)_X \\ &\quad= \mathrm{CYEF}(Z)(y2x) \star \big(\mathrm{FF}(\alpha_{f_Z,X}) \star \mu_{\mathrm{EF}(X)}\big) \;=\; \big(\mathrm{CYEF}(Z)(y2x) \star \mathrm{FF}(\alpha_{f_Z,X})\big) \star \mu_{\mathrm{EF}(X)} \\ &\quad= \mathrm{FF}\big(z2y \mapsto f_Z \star (G \circ \mathrm{EF})(z2y \star y2x)\big) \star \mu_{\mathrm{EF}(X)} \\ &\quad= \mathrm{FF}\big(z2y \mapsto (f_Z \star (G \circ \mathrm{EF})(z2y)) \star (G \circ \mathrm{EF})(y2x)\big) \star \mu_{\mathrm{EF}(X)} \\ &\quad= \big(\mathrm{FF}(\alpha_{f_Z,Y}) \star (G \circ G)(\mathrm{EF}(y2x))\big) \star \mu_{\mathrm{EF}(X)} \;=\; \mathrm{FF}(\alpha_{f_Z,Y}) \star \big((G \circ G)(\mathrm{EF}(y2x)) \star \mu_{\mathrm{EF}(X)}\big) \\ &\quad= \mathrm{FF}(\alpha_{f_Z,Y}) \star \big(\mu_{\mathrm{EF}(Y)} \star (G \circ \mathrm{EF})(y2x)\big) \\ &\quad= \big(\mathrm{FF}(\alpha_{f_Z,Y}) \star \mu_{\mathrm{EF}(Y)}\big) \star (G \circ \mathrm{EF})(y2x) \;=\; (f_Z \to \tau_Z)_Y \star (G \circ \mathrm{EF})(y2x). \end{aligned}$$

Part (ii): Section. Let $f_Z : \mathrm{FF}(T) \to G(\mathrm{EF}(Z))$. Using $(G \circ \mathrm{EF})(\mathrm{id}_Z) = \mathrm{id}_{G(\mathrm{EF}(Z))}$, Lemma 2.3 ($\mathrm{id}_{\mathrm{FF}(T)} = g_T(t)$ for $t \in T$), functoriality of FF on g_T followed by $(h \mapsto h \star f_Z)$, the parametric unit law (3) at $Z = T$ ($\mathrm{FF}(g_T) = \eta_{\mathrm{FF}(T)}$), naturality of η on f_Z, and the left identity law (5):

$$\begin{aligned} ((f_Z \to \tau_Z) \to f_Z) &= \iota_Z \star \big(\mathrm{FF}(z2z \mapsto f_Z \star (G \circ \mathrm{EF})(z2z)) \star \mu_{\mathrm{EF}(Z)}\big) \\ &= \big(g(\mathrm{id}_Z) \star \mathrm{FF}(z2z \mapsto f_Z \star (G \circ \mathrm{EF})(z2z))\big) \star \mu_{\mathrm{EF}(Z)} \\ &= g\big(f_Z \star (G \circ \mathrm{EF})(\mathrm{id}_Z)\big) \star \mu_{\mathrm{EF}(Z)} \\ &= g(f_Z \star \mathrm{id}_{G(\mathrm{EF}(Z))}) \star \mu_{\mathrm{EF}(Z)} \;=\; g(f_Z) \star \mu_{\mathrm{EF}(Z)} \;=\; g(\mathrm{id}_{\mathrm{FF}(T)} \star f_Z) \star \mu_{\mathrm{EF}(Z)} \\ &= \mathrm{FF}\big(t \mapsto g_T(t) \star f_Z\big) \star \mu_{\mathrm{EF}(Z)} \\ &= \big(\mathrm{FF}(g_T) \star G(f_Z)\big) \star \mu_{\mathrm{EF}(Z)} \\ &= \mathrm{FF}(g_T) \star \big(G(f_Z) \star \mu_{\mathrm{EF}(Z)}\big) \\ &= \eta_{\mathrm{FF}(T)} \star \big(G(f_Z) \star \mu_{\mathrm{EF}(Z)}\big) \\ &= \big(\eta_{\mathrm{FF}(T)} \star G(f_Z)\big) \star \mu_{\mathrm{EF}(Z)} \\ &= \big(f_Z \star \eta_{G(\mathrm{EF}(Z))}\big) \star \mu_{\mathrm{EF}(Z)} \\ &= f_Z \star \big(\eta_{G(\mathrm{EF}(Z))} \star \mu_{\mathrm{EF}(Z)}\big) \;=\; f_Z \star \mathrm{id}_{G(\mathrm{EF}(Z))} \;=\; f_Z. \end{aligned}$$

Part (iii): Retraction. Let $\tau_Z : \mathrm{CYEF}(Z) \Rightarrow G \circ \mathrm{EF}$ and $Y \in \mathrm{ob}(\mathcal{C})$. Applying associativity, naturality of τ_Z along $z2y : Z \to Y$, the identity $\iota_Z \star \mathrm{CYEF}(Z)(z2y) = g(\mathrm{id}_Z \star z2y) = g(z2y)$, the parametric unit law (3) at the hom-set $H_{Z,Y} \coloneqq \mathrm{Hom}_{\mathcal{C}}(Z, Y)$ ($\mathrm{FF}(g_{H_{Z,Y}}) = \eta_{\mathrm{FF}(\mathrm{Hom}_{\mathcal{C}}(Z,Y))} = \eta_{\mathrm{CYEF}(Z)(Y)}$), naturality of η along $(\tau_Z)_Y$, and the left identity law (5):

$$\begin{aligned}
((\tau_Z \to f_Z) \to \tau_Z)_Y &= \mathrm{FF}\big(z2y \mapsto (\iota_Z \star (\tau_Z)_Z) \star (G \circ \mathrm{EF})(z2y)\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}\big(z2y \mapsto \iota_Z \star ((\tau_Z)_Z \star (G \circ \mathrm{EF})(z2y))\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}\big(z2y \mapsto \iota_Z \star (\mathrm{CYEF}(Z)(z2y) \star (\tau_Z)_Y)\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}\big(z2y \mapsto (\iota_Z \star \mathrm{CYEF}(Z)(z2y)) \star (\tau_Z)_Y\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}\big(z2y \mapsto g(\mathrm{id}_Z \star z2y) \star (\tau_Z)_Y\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}\big(z2y \mapsto g(z2y) \star (\tau_Z)_Y\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \big(\mathrm{FF}(g_{H_{Z,Y}}) \star G((\tau_Z)_Y)\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \mathrm{FF}(g_{H_{Z,Y}}) \star \big(G((\tau_Z)_Y) \star \mu_{\mathrm{EF}(Y)}\big) \\
&= \eta_{\mathrm{CYEF}(Z)(Y)} \star \big(G((\tau_Z)_Y) \star \mu_{\mathrm{EF}(Y)}\big) \\
&= \big(\eta_{\mathrm{CYEF}(Z)(Y)} \star G((\tau_Z)_Y)\big) \star \mu_{\mathrm{EF}(Y)} \\
&= \big((\tau_Z)_Y \star \eta_{G(\mathrm{EF}(Y))}\big) \star \mu_{\mathrm{EF}(Y)} \\
&= (\tau_Z)_Y \star \big(\eta_{G(\mathrm{EF}(Y))} \star \mu_{\mathrm{EF}(Y)}\big) \;=\; (\tau_Z)_Y \star \mathrm{id}_{G(\mathrm{EF}(Y))} \;=\; (\tau_Z)_Y.
\end{aligned}$$

By extensionality of natural transformations, $((\tau_Z \to f_Z) \to \tau_Z) = \tau_Z$. □

3.2 Connecting the CoEndoYoneda Isomorphism to η at $\mathrm{FF}(T)$

When we specialize Theorem 3.4 to $\mathrm{EF} = \mathrm{Id}_{\mathcal{C}}$ and $Z = \mathrm{FF}(T)$, the domain functor of the left-hand side becomes $\mathrm{CYEF}(\mathrm{FF}(T)) = G$, and the codomain functor becomes $G \circ \mathrm{Id}_{\mathcal{C}} = G$. Hence the left-hand side contains the canonical identity natural transformation $\mathrm{id}_G : G \Rightarrow G$.

Proposition 3.5. *Define the morphism*

$$\eta^{\mathrm{CY}}_{\mathrm{FF}(T)} \;\coloneqq\; (\mathrm{id}_G \to f_{\mathrm{FF}(T)}) \;:\; \mathrm{FF}(T) \longrightarrow G(\mathrm{FF}(T))$$

as the image of id_G *under the forward map* (7) *of the CoEndoYoneda isomorphism at* $\mathrm{EF} = \mathrm{Id}_{\mathcal{C}}$ *and* $Z = \mathrm{FF}(T)$*. Then in any Basic Functional Category,*

$$\eta^{\mathrm{CY}}_{\mathrm{FF}(T)} \;=\; \eta_{\mathrm{FF}(T)}.$$

Proof. Unfolding (7) at $\tau = \mathrm{id}_G$, and applying right identity, Lemma 2.3, and Proposition 2.5:

$$\eta^{\mathrm{CY}}_{\mathrm{FF}(T)} \;=\; \iota_{\mathrm{FF}(T)} \star \mathrm{id}_{G(\mathrm{FF}(T))} \;=\; g(\mathrm{id}_{\mathrm{FF}(T)}) \;=\; g(g(\mathrm{tt}^*)) \;=\; \eta_{\mathrm{FF}(T)}.$$

□

3.3 Canonical Concrete Instance: SET as a Functional Category

Proposition 3.6. *In the category* $\mathcal{C} = \mathbf{SET}_\ell$ *with* $\mathrm{FF} = \mathrm{Id}_{\mathbf{SET}_\ell}$*, we have* $G(Y) = \mathrm{CYEF}(T)(Y) = (T \to Y) \cong Y$*. Define for each* $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$*:*

$$\eta_Z(z) \;\coloneqq\; \mathrm{const}_z \;:\; T \to Z, \qquad \mu_Z(h) \;\coloneqq\; (t \mapsto h(t)(t)) \;:\; T \to Z \quad \text{for } h : T \to (T \to Z).$$

Then $\eta : \mathrm{Id}_{\mathbf{SET}_\ell} \Rightarrow G$ *and* $\mu : G \circ G \Rightarrow G$ *are natural transformations, and the laws* (3), (4), *and* (5) *all hold definitionally.*

Proof. Since $\mathrm{FF} = \mathrm{Id}_{\mathbf{SET}_\ell}$, $\mathrm{FF}(g_Z)(z) = g_Z(z) = \mathrm{const}_z = \eta_Z(z)$ definitionally, proving (3) (and hence (4)) by reflexivity. For any $h : T \to Z$ and $t \in T$, $(\eta_{G(Z)} \star \mu_Z)(h)(t) = \mu_Z(_ \mapsto h)(t) = (_ \mapsto h)(t)(t) = h(t)$, so $\eta_{G(Z)} \star \mu_Z = \mathrm{id}_{G(Z)}$ holds definitionally by $\beta\eta$-reduction. □

3.4 Relationship to the Classical CoYoneda Lemma: Naturality, Universality, Generalization, and Specialization

It is illuminating to compare the CoEndoYoneda Isomorphism Theorem (Theorem 3.4) with the classical CoYoneda lemma (1) across four foundational dimensions:

(1) **Endofunctor coincidence on $\mathcal{C} = \mathbf{SET}$ and generalization to $(\mathcal{C}, \mathrm{FF})$:** In the classical CoYoneda lemma $\mathrm{NatTrans}(\mathrm{CYF}(Z), F) \cong F(Z)$, the domain functor $\mathrm{CYF}(Z) = \mathrm{Hom}_{\mathcal{C}}(Z, -) : \mathcal{C} \to \mathbf{SET}$ is always $\mathbf{SET}$-valued. In the special case where the base category $\mathcal{C}$ is $\mathbf{SET}$ itself, $\mathrm{CYF}(Z) : \mathbf{SET} \to \mathbf{SET}$ and $F = \mathrm{EF} : \mathbf{SET} \to \mathbf{SET}$ are both *endofunctors* on $\mathbf{SET}$. When we instantiate our CoEndoYoneda Isomorphism Theorem 3.4 at the canonical Functional Category $(\mathcal{C}, \mathrm{FF}) = (\mathbf{SET}, \mathrm{Id}_{\mathbf{SET}})$ of Proposition 3.6, we have $\mathrm{CYEF}(Z) = \mathrm{Id}_{\mathbf{SET}} \circ \mathrm{CYF}(Z) = \mathrm{CYF}(Z)$, $G = (T \to -) \cong \mathrm{Id}_{\mathbf{SET}}$, and

$$\mathrm{NatTrans}(\mathrm{CYEF}(Z),\, G{\circ}\mathrm{EF}) \;\cong\; \mathrm{Hom}_{\mathbf{SET}}(T,\, G(\mathrm{EF}(Z))) \;=\; (T \to (T \to \mathrm{EF}(Z))) \;\cong\; \mathrm{EF}(Z),$$

which recovers the classical CoYoneda lemma for endofunctors on $\mathbf{SET}$. Conversely, for an arbitrary Functional Category $(\mathcal{C}, \mathrm{FF})$, Theorem 3.4 *generalizes* this endofunctor CoYoneda isomorphism from $\mathbf{SET}$ to endofunctors on $\mathcal{C}$ by lifting $\mathrm{CYF}(Z)$ to $\mathrm{CYEF}(Z) = \mathrm{FF} \circ \mathrm{CYF}(Z) : \mathcal{C} \to \mathcal{C}$ and replacing set-theoretic elements of $F(Z)$ by global-like morphisms $\mathrm{FF}(T) \to G(\mathrm{EF}(Z))$ in $\mathcal{C}$.

(2) **Pointfree universality and the universal morphism ι_Z:** In the classical CoYoneda lemma, the pair $(Z, \mathrm{id}_Z \in \mathrm{CYF}(Z)(Z))$ is the universal element of the representable presheaf $\mathrm{CYF}(Z)$: every natural transformation $\tau : \mathrm{CYF}(Z) \Rightarrow F$ is uniquely determined by the single element $x = \tau_Z(\mathrm{id}_Z) \in F(Z)$. In a general category $\mathcal{C}$, however, the object $\mathrm{CYEF}(Z)(Z) = \mathrm{FF}(\mathrm{Hom}_{\mathcal{C}}(Z, Z)) \in \mathrm{ob}(\mathcal{C})$ is an abstract object of $\mathcal{C}$ rather than a set, so it has no "elements" in the pointful sense. In Theorem 3.4, the role of the universal element id_Z is played pointfreely by the universal global-element morphism

$$\iota_Z \;=\; g_{\mathrm{Hom}_{\mathcal{C}}(Z,Z)}(\mathrm{id}_Z) \;=\; \mathrm{FF}(\mathrm{const}_{\mathrm{id}_Z}) \;:\; \mathrm{FF}(T) \longrightarrow \mathrm{CYEF}(Z)(Z) \quad \text{in } \mathcal{C},$$

and evaluation at id_Z is replaced by diagrammatic precomposition $\tau_Z \mapsto \iota_Z \star (\tau_Z)_Z$. Moreover, Proposition 3.5 shows that the universal element corresponding to the identity natural transformation $\mathrm{id}_G : G \Rightarrow G$ at $Z = \mathrm{FF}(T)$ is precisely the terminal unit component $\eta_{\mathrm{FF}(T)} = \iota_{\mathrm{FF}(T)} \star \mathrm{id}_{G(\mathrm{FF}(T))} = g(\mathrm{id}_{\mathrm{FF}(T)})$.

(3) **Role of the Functional Category structure (η, μ) in reconstruction:** In the classical CoYoneda lemma, reconstructing $\tau_Y : \mathrm{Hom}_{\mathcal{C}}(Z, Y) \to F(Y)$ from $x \in F(Z)$ sends $z2y \mapsto F(z2y)(x)$. In our setting, given $f_Z : \mathrm{FF}(T) \to G(\mathrm{EF}(Z))$, postcomposing with $(G \circ \mathrm{EF})(z2y)$ yields a function $\alpha_{f_Z,Y} : \mathrm{Hom}_{\mathcal{C}}(Z, Y) \to \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), G(\mathrm{EF}(Y)))$ in $\mathbf{SET}$. Applying FF produces a morphism

$$\mathrm{FF}(\alpha_{f_Z,Y}) \;:\; \mathrm{CYEF}(Z)(Y) \longrightarrow G\big(G(\mathrm{EF}(Y))\big) \quad \text{in } \mathcal{C},$$

which lands in $G(G(\mathrm{EF}(Y)))$ with *two* layers of G. The multiplication-like transformation $\mu_{\mathrm{EF}(Y)} : G(G(\mathrm{EF}(Y))) \to G(\mathrm{EF}(Y))$ flattens these two layers into one, while the parametric unit law $\eta_{\mathrm{FF}(Z)} = \mathrm{FF}(g_Z)$ and the left unit law $\eta_{G(X)} \star \mu_X = \mathrm{id}_{G(X)}$ are the exact coherence axioms needed to guarantee that flattening after lifting is a two-sided inverse to precomposition with ι_Z.

(4) **Comparison with the adjunction perspective:** When $\mathrm{FF} : \mathbf{SET} \to \mathcal{C}$ is a left adjoint to the global-elements functor $\mathrm{CYF}(\mathrm{FF}(T)) = \mathrm{Hom}_{\mathcal{C}}(\mathrm{FF}(T), -) : \mathcal{C} \to \mathbf{SET}$, natural transformations $\mathrm{FF}{\circ}\mathrm{CYF}(Z) \Rightarrow G{\circ}\mathrm{EF}$ transpose across the adjunction to natural transformations $\mathrm{CYF}(Z) \Rightarrow \mathrm{CYF}(\mathrm{FF}(T)) \circ G \circ \mathrm{EF}$ in $\mathbf{SET}$, where the classical CoYoneda lemma applies. Crucially, however, a Functional Category $(\mathcal{C}, \mathrm{FF})$ *does not assume an*

adjunction $\mathrm{FF} \dashv \mathrm{CYF}(\mathrm{FF}(T))$ (nor even that G is a full monad, since neither the right unit law nor associativity of μ is required). Instead, Theorem 3.4 isolates the minimal pointfree equational core—Equations (3) and (5)—under which the CoEndoYoneda bijection holds directly inside $\mathcal{C}$.

4 Purity, Idempotent Monads, and Kleisli Categories

We now turn to the connection between Basic Functional Categories and computational purity for monads on $\mathbf{SET}_\ell$.

4.1 Monad Formulations and Characterizations of Purity

In functional programming, a monad $(M, \mathrm{mPure}, \mathrm{mApply})$ on $\mathbf{SET}_\ell$ is given by an object mapping $M : \mathrm{ob}(\mathbf{SET}_\ell) \to \mathrm{ob}(\mathbf{SET}_\ell)$, a unit family $\mathrm{mPure}_Z : Z \to M(Z)$, and a Kleisli extension (bind) operation $\mathrm{mApply} : (Z \to M(Y)) \to M(Z) \to M(Y)$ satisfying the three monad laws:

$$\mathrm{mApply}\ f\ (\mathrm{mPure}_Z\ z) = f(z) \qquad \text{(left unit law)}, \tag{9}$$

$$\mathrm{mApply}\ \mathrm{mPure}_Z\ mz = mz \qquad \text{(right unit law)}, \tag{10}$$

$$\mathrm{mApply}\ h\ (\mathrm{mApply}\ f\ mz) = \mathrm{mApply}\ (z \mapsto \mathrm{mApply}\ h\ (f(z)))\ mz \qquad \text{(associativity law)}. \tag{11}$$

From these, one defines the functorial map operation mMap (for $f : Z \to Y$) and monadic multiplication mJoin:

$$\mathrm{mMap}\ f\ mz \coloneqq \mathrm{mApply}\ (\mathrm{mPure}_Y \circ f)\ mz, \qquad \mathrm{mJoin}\ mmz \coloneqq \mathrm{mApply}\ \mathrm{id}_{M(Z)}\ mmz, \tag{12}$$

which satisfy all standard categorical monad and naturality laws, with $\mathrm{mApply}\ f\ mz = \mathrm{mJoin}(\mathrm{mMap}\ f\ mz)$.

Definition 4.1. Let $(M, \mathrm{mPure}, \mathrm{mApply})$ be a monad on $\mathbf{SET}_\ell$.

(i) M is *idempotent* ($\mathrm{Idempotent}(M)$, cfr. $M\eta = \eta M$ in [1, 11]) if for every $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ and $mz \in M(Z)$,

$$\mathrm{mMap}\ \mathrm{mPure}_Z\ mz\ =\ \mathrm{mPure}_{M(Z)}\ mz. \tag{13}$$

Intuitively, performing an effect twice is the same as performing it once.

(ii) M *enforces purity* ($\mathrm{IsPure}(M)$) if for every $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ and $mz \in M(Z)$,

$$\mathrm{mApply}\ (\mathrm{mPure}_{M(Z)} \circ \mathrm{mPure}_Z)\ mz\ =\ \mathrm{mPure}_{M(Z)}\ mz. \tag{14}$$

Lemma 4.2. *For any monad M on $\mathbf{SET}_\ell$,* $\mathrm{Idempotent}(M) \iff \mathrm{IsPure}(M)$.

Proof. Immediate because $\mathrm{mMap}\ \mathrm{mPure}_Z\ mz = \mathrm{mApply}\ (\mathrm{mPure}_{M(Z)} \circ \mathrm{mPure}_Z)\ mz$ by definition (12) of mMap. □

4.2 Equivalence Between Kleisli Basic Functional Categories and Monad Idempotency

Given a monad M on $\mathbf{SET}_\ell$, its *Kleisli category* $\mathbf{K}(M)$ has:

- **Objects:** $\mathrm{ob}(\mathbf{K}(M)) \coloneqq \mathrm{ob}(\mathbf{SET}_\ell)$;
- **Morphisms:** $\mathrm{Hom}_{\mathbf{K}(M)}(Z, Y) \coloneqq (Z \to M(Y))$;
- **Identities:** $\mathrm{id}_Z^{\mathbf{K}(M)} \coloneqq \mathrm{mPure}_Z : Z \to M(Z)$;

- **Kleisli composition** (denoted $\star_{\mathbf{K}}$ to distinguish it from ordinary function composition in $\mathbf{SET}_\ell$): for $f \in \mathrm{Hom}_{\mathbf{K}(M)}(Z, Y)$ and $h \in \mathrm{Hom}_{\mathbf{K}(M)}(Y, X)$,

$$(f \star_{\mathbf{K}} h)(z) \;\coloneqq\; \mathrm{mApply}\ h\ (f(z)) \;\in\; M(X); \tag{15}$$

- **Canonical functional functor** $\mathrm{KFF} : \mathbf{SET}_\ell \to \mathbf{K}(M)$ (identity on objects):

$$\mathrm{KFF}(Z) \;\coloneqq\; Z, \qquad \mathrm{KFF}(f : Z \to Y) \;\coloneqq\; \big(z \mapsto \mathrm{mPure}_Y(f(z))\big) \;\in\; \mathrm{Hom}_{\mathbf{K}(M)}(Z, Y). \tag{16}$$

In $(\mathbf{K}(M), \mathrm{KFF})$, let us explicitly unpack the global-like endofunctor $G = \mathrm{CYEF}(\mathrm{KFF}(T)) = \mathrm{KFF} \circ \mathrm{CYF}(T) : \mathbf{K}(M) \to \mathbf{K}(M)$ and the global-element generator g_Z:

- For any object $Z \in \mathrm{ob}(\mathbf{K}(M))$, since KFF is the identity on objects,

$$G(Z) \;=\; \mathrm{KFF}\big(\mathrm{Hom}_{\mathbf{K}(M)}(T, Z)\big) \;=\; \mathrm{Hom}_{\mathbf{K}(M)}(T, Z) \;=\; \big(T \to M(Z)\big).$$

- For any Kleisli morphism $f : Y \to M(X)$ in $\mathrm{Hom}_{\mathbf{K}(M)}(Y, X)$, the function $\mathrm{CYF}(T)(f) : (T \to M(Y)) \to (T \to M(X))$ maps $g_{MY} : T \to M(Y)$ to the Kleisli composite $g_{MY} \star_{\mathbf{K}} f = \big(t \mapsto \mathrm{mApply}\ f\ (g_{MY}(t))\big)$. Applying KFF according to (16) yields

$$G(f)(g_{MY}) \;=\; \mathrm{mPure}_{G(X)}\big(t \mapsto \mathrm{mApply}\ f\ (g_{MY}(t))\big) \;\in\; M\big(T \to M(X)\big). \tag{17}$$

- For any set Z and element $z \in Z$, the global-element generator $g_Z(z) = \mathrm{KFF}(\mathrm{const}_z) \in \mathrm{Hom}_{\mathbf{K}(M)}(T, Z)$ is the constant function $g_Z(z) = \big(_ \mapsto \mathrm{mPure}_Z(z)\big) : T \to M(Z)$. Consequently, lifting $g_Z : Z \to G(Z)$ via KFF yields

$$\mathrm{KFF}(g_Z)(z) \;=\; \mathrm{mPure}_{G(Z)}\big(g_Z(z)\big) \;=\; \mathrm{mPure}_{G(Z)}\big(_ \mapsto \mathrm{mPure}_Z(z)\big) \;\in\; M\big(T \to M(Z)\big), \tag{18}$$

  and at $Z = T$ with $z = \mathrm{tt}^*$, the terminal law (4) states that

$$\eta_T(\mathrm{tt}^*) \;=\; g_{G(T)}\big(g_T(\mathrm{tt}^*)\big)(\mathrm{tt}^*) \;=\; \mathrm{mPure}_{G(T)}\big(_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*)\big). \tag{19}$$

Theorem 4.3. *Let* $(M, \mathrm{mPure}, \mathrm{mApply})$ *be a monad on* $\mathbf{SET}_\ell$*. Then* $(\mathbf{K}(M), \mathrm{KFF})$ *is a* Basic Functional Category *if and only if* M *is an* Idempotent *(pure) monad:*

$$\mathrm{BasicFunctionalCategory}\big(\mathbf{K}(M),\, \mathrm{KFF}\big) \iff \mathrm{Idempotent}(M).$$

More specifically, there is a striking asymmetry between the two directions:

(i) ***(⟹):*** *Only naturality of* η *and the terminal specialization* (4) *(*$\eta_T = g_{G(T)}(g_T(\mathrm{tt}^*))$*) are used to deduce that* M *is globally idempotent on all sets* Z*.*

(ii) ***(⟸):*** *Assuming* M *is idempotent, the candidate unit* $\eta_Y(y) \coloneqq \mathrm{mPure}_{G(Y)}(_ \mapsto \mathrm{mPure}_Y(y))$ *is natural and satisfies the full parametric law* (3) *(*$\eta_Z = \mathrm{KFF}(g_Z)$*) definitionally by reflexivity.*

Proof. We provide the complete, self-contained equational proof of both directions, matching the mechanized derivations in [3, 4].

Direction (⟹): $\mathrm{BasicFunctionalCategory}(\mathbf{K}(M), \mathrm{KFF}) \implies \mathrm{Idempotent}(M)$**.** Suppose $(\mathbf{K}(M), \mathrm{KFF})$ is a Basic Functional Category with unit natural transformation $\eta : \mathrm{Id}_{\mathbf{K}(M)} \Rightarrow G$, where for each set Z, $\eta_Z : Z \to M(T \to M(Z))$ is a Kleisli morphism in $\mathrm{Hom}_{\mathbf{K}(M)}(Z, G(Z))$.

Step 1: Proof of the Evaluation Lemma. We first prove that for every set $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ and every monadic computation $mz \in M(Z)$,

$$\mathrm{mApply}\ \eta_Z\ mz \;=\; \mathrm{mPure}_{G(Z)}(_ \mapsto mz) \;\in\; M\big(T \to M(Z)\big). \tag{20}$$

Define the constant Kleisli morphism $t \to mz \in \mathrm{Hom}_{\mathbf{K}(M)}(T, Z)$ by

$$(t \to mz)(t) \;\coloneqq\; mz \qquad \text{for all } t \in T.$$

By naturality of $\eta : \mathrm{Id}_{\mathbf{K}(M)} \Rightarrow G$ applied to the Kleisli morphism $t \to mz : T \to Z$ in $\mathbf{K}(M)$, we have the equality of Kleisli morphisms in $\mathrm{Hom}_{\mathbf{K}(M)}(T, G(Z))$:

$$(t \to mz) \star_{\mathbf{K}} \eta_Z \;=\; \eta_T \star_{\mathbf{K}} G(t \to mz). \tag{21}$$

We evaluate both sides of (21) at the unique element $\mathrm{tt}^* \in T$. By definition (15) of Kleisli composition, the left-hand side of (21) evaluated at tt^* is:

$$\big((t \to mz) \star_{\mathbf{K}} \eta_Z\big)(\mathrm{tt}^*) \;=\; \mathrm{mApply}\ \eta_Z\ \big((t \to mz)(\mathrm{tt}^*)\big) \;=\; \mathrm{mApply}\ \eta_Z\ mz. \tag{22}$$

On the right-hand side of (21), evaluating at tt^* and applying the Kleisli composition formula (15), the terminal unit law (19), the left unit monad law (9), the action of G in (17), and a second application of the left unit monad law (9) yields:

$$\begin{aligned}
&\big(\eta_T \star_{\mathbf{K}} G(t \to mz)\big)(\mathrm{tt}^*) \\
&\quad= \mathrm{mApply}\ \big(G(t \to mz)\big)\ \big(\eta_T(\mathrm{tt}^*)\big) && \text{(by (15))} \\
&\quad= \mathrm{mApply}\ \big(G(t \to mz)\big)\ \big(\mathrm{mPure}_{G(T)}(_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*))\big) && \text{(by (19))} \\
&\quad= G(t \to mz)\big(_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*)\big) && \text{(by (9))} \\
&\quad= \mathrm{mPure}_{G(Z)}\big(t \mapsto \mathrm{mApply}\ (t \to mz)\ ((_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*))(t))\big) && \text{(by (17))} \\
&\quad= \mathrm{mPure}_{G(Z)}\big(t \mapsto \mathrm{mApply}\ (t \to mz)\ (\mathrm{mPure}_T(\mathrm{tt}^*))\big) && (\beta\text{-red.}) \\
&\quad= \mathrm{mPure}_{G(Z)}\big(t \mapsto (t \to mz)(\mathrm{tt}^*)\big) && \text{(by (9))} \\
&\quad= \mathrm{mPure}_{G(Z)}(_ \mapsto mz) && \text{(def. } t \to mz\text{).}
\end{aligned} \tag{23}$$

Combining (21), (22), and (23) establishes the Evaluation Lemma (20).

Step 2: Point Evaluation Homomorphism and Pointwise Idempotency Lemma. For each set Z, define the evaluation map $\mathrm{mgz} \to \mathrm{mmz} : M(G(Z)) \to M(M(Z))$ by

$$\mathrm{mgz} \to \mathrm{mmz}(m_{t \to mz}) \;\coloneqq\; \mathrm{mApply}\ \big(t2mz \mapsto \mathrm{mPure}_{M(Z)}(t2mz(\mathrm{tt}^*))\big)\ m_{t \to mz}. \tag{24}$$

Note that for any function $h : T \to M(Z)$, the left unit monad law (9) gives

$$\begin{aligned}
\mathrm{mgz} \to \mathrm{mmz}\big(\mathrm{mPure}_{G(Z)}(h)\big) &= \mathrm{mApply}\ \big(t2mz \mapsto \mathrm{mPure}_{M(Z)}(t2mz(\mathrm{tt}^*))\big)\ \big(\mathrm{mPure}_{G(Z)}(h)\big) \\
&= \mathrm{mPure}_{M(Z)}\big(h(\mathrm{tt}^*)\big).
\end{aligned} \tag{25}$$

Now, for any element $z \in Z$, applying the left unit monad law (9) to write $\eta_Z(z) = \mathrm{mApply}\ \eta_Z\ (\mathrm{mPure}_Z(z))$, then applying the Evaluation Lemma (20) to $mz = \mathrm{mPure}_Z(z)$, and finally applying (25) to $h = (_ \mapsto \mathrm{mPure}_Z(z))$ gives:

$$\begin{aligned}
\mathrm{mgz} \to \mathrm{mmz}\big(\eta_Z(z)\big) &= \mathrm{mgz} \to \mathrm{mmz}\Big(\mathrm{mApply}\ \eta_Z\ \big(\mathrm{mPure}_Z(z)\big)\Big) && \text{(by (9))} \\
&= \mathrm{mgz} \to \mathrm{mmz}\Big(\mathrm{mPure}_{G(Z)}\big(_ \mapsto \mathrm{mPure}_Z(z)\big)\Big) && \text{(by (20))} \\
&= \mathrm{mPure}_{M(Z)}\big(\mathrm{mPure}_Z(z)\big) && \text{(by (25)).}
\end{aligned} \tag{26}$$

Step 3: Global Idempotency of M. For any set Z and any computation $mz \in M(Z)$, notice first that for any $m_{t \to mz} = \mathrm{mApply}\ \eta_Z\ mz$, the monad associativity law (11) and definition (24) yield

$$\begin{aligned}
&\mathrm{mgz} \to \mathrm{mmz}(\mathrm{mApply}\ \eta_Z\ mz) \\
&\quad= \mathrm{mApply}\ \big(t2mz \mapsto \mathrm{mPure}_{M(Z)}(t2mz(\mathrm{tt}^*))\big)\ (\mathrm{mApply}\ \eta_Z\ mz) \\
&\quad= \mathrm{mApply}\ \big(z \mapsto \mathrm{mApply}\ (t2mz \mapsto \mathrm{mPure}_{M(Z)}(t2mz(\mathrm{tt}^*)))\ (\eta_Z(z))\big)\ mz \\
&\quad= \mathrm{mApply}\ \big(z \mapsto \mathrm{mgz} \to \mathrm{mmz}(\eta_Z(z))\big)\ mz.
\end{aligned} \tag{27}$$

Therefore, we obtain the complete equational chain:

$$\begin{aligned}
\mathrm{mMap}\ \mathrm{mPure}_Z\ mz &= \mathrm{mApply}\ (z \mapsto \mathrm{mPure}_{M(Z)}(\mathrm{mPure}_Z(z)))\ mz && \text{(by (12))}\\
&= \mathrm{mApply}\ (z \mapsto \mathrm{mgz}{\to}\mathrm{mmz}(\eta_Z(z)))\ mz && \text{(by (26) and ext.)}\\
&= \mathrm{mgz}{\to}\mathrm{mmz}(\mathrm{mApply}\ \eta_Z\ mz) && \text{(by (27))}\\
&= \mathrm{mgz}{\to}\mathrm{mmz}(\mathrm{mPure}_{G(Z)}(_ \mapsto mz)) && \text{(by (20))}\\
&= \mathrm{mPure}_{M(Z)}(mz) && \text{(by (25), } h = (_ \mapsto mz)).
\end{aligned}$$

Thus $\mathrm{mMap}\ \mathrm{mPure}_Z\ mz = \mathrm{mPure}_{M(Z)}(mz)$ for all Z and $mz \in M(Z)$, proving $\mathrm{Idempotent}(M)$.

Direction ($\Longleftarrow$): $\mathrm{Idempotent}(M) \implies \mathrm{BasicFunctionalCategory}(\mathbf{K}(M), \mathrm{KFF})$. Conversely, assume $\mathrm{Idempotent}(M)$ holds. For every set $Z \in \mathrm{ob}(\mathbf{SET}_\ell)$ and $mz \in M(Z)$, define $mz \to mgmz := (m \mapsto \mathrm{mPure}_{G(Z)}(_ \mapsto m)) : M(Z) \to M(G(Z))$. By the left unit law (9), for each $z \in Z$ we have $\mathrm{mPure}_{G(Z)}(_ \mapsto \mathrm{mPure}_Z(z)) = \mathrm{mApply}\ mz{\to}mgmz\ (\mathrm{mPure}_{M(Z)}(\mathrm{mPure}_Z(z)))$. Applying the associativity law (11), definition (12) of mMap, the idempotency hypothesis (13), and the left unit law (9) gives:

$$\begin{aligned}
&\mathrm{mApply}\ (z \mapsto \mathrm{mPure}_{G(Z)}(_ \mapsto \mathrm{mPure}_Z(z)))\ mz\\
&\quad= \mathrm{mApply}\ (z \mapsto \mathrm{mApply}\ mz{\to}mgmz\ (\mathrm{mPure}_{M(Z)}(\mathrm{mPure}_Z(z))))\ mz\\
&\quad= \mathrm{mApply}\ mz{\to}mgmz\ (\mathrm{mApply}\ (z \mapsto \mathrm{mPure}_{M(Z)}(\mathrm{mPure}_Z(z)))\ mz)\\
&\quad= \mathrm{mApply}\ mz{\to}mgmz\ (\mathrm{mMap}\ \mathrm{mPure}_Z\ mz)\\
&\quad= \mathrm{mApply}\ mz{\to}mgmz\ (\mathrm{mPure}_{M(Z)}(mz)) \ =\ \mathrm{mPure}_{G(Z)}(_ \mapsto mz). \qquad (28)
\end{aligned}$$

Now define the candidate unit $\eta_Y : Y \to M(G(Y))$ for each $Y \in \mathrm{ob}(\mathbf{K}(M))$ by

$$\eta_Y(y) \ :=\ \mathrm{mPure}_{G(Y)}(_ \mapsto \mathrm{mPure}_Y(y)).$$

For any Kleisli morphism $y \to mx : Y \to M(X)$ and any $y \in Y$, applying (15), (28) at $mz = y{\to}mx\ (y)$, the left unit law (9) twice, and (17) yields naturality:

$$\begin{aligned}
&((y{\to}mx) \star_{\mathbf{K}} \eta_X)(y)\\
&\quad= \mathrm{mApply}\ (x \mapsto \mathrm{mPure}_{G(X)}(_ \mapsto \mathrm{mPure}_X(x)))\ (y{\to}mx\ (y))\\
&\quad= \mathrm{mPure}_{G(X)}(_ \mapsto y{\to}mx\ (y)) \ =\ \mathrm{mPure}_{G(X)}(_ \mapsto \mathrm{mApply}\ y{\to}mx\ (\mathrm{mPure}_Y(y)))\\
&\quad= \mathrm{mApply}\ (gmy \mapsto \mathrm{mPure}_{G(X)}(t \mapsto \mathrm{mApply}\ y{\to}mx\ (gmy(t))))\ (\mathrm{mPure}_{G(Y)}(_ \mapsto \mathrm{mPure}_Y(y)))\\
&\quad= (\eta_Y \star_{\mathbf{K}} G(y{\to}mx))(y).
\end{aligned}$$

Finally, by (18), $\eta_Z(z) = \mathrm{mPure}_{G(Z)}(_ \mapsto \mathrm{mPure}_Z(z)) = \mathrm{KFF}(g_Z)(z)$ holds definitionally by reflexivity for every set Z, establishing (3). □

4.3 The Terminal Subsingleton Theorem

A classical property of idempotent monads (building on Lawvere and Manes, with the equational proof presented in Borceux [1]) is that any monadic computation over the terminal type T must be equal to $\mathrm{mPure}_T(\mathrm{tt}^*)$:

Theorem 4.4. *Let M be a monad on $\mathbf{SET}_\ell$ satisfying $\mathrm{IsPure}(M)$. Then M satisfies the terminal subsingleton property $\mathrm{MonadicTerminalTypeSubSingleton}(M)$: for every terminal monadic computation $mt \in M(T)$,*

$$mt \ =\ \mathrm{mPure}_T(\mathrm{tt}^*). \qquad (29)$$

In particular, $M(T)$ is a subsingleton ($\forall m_1, m_2 \in M(T),\ m_1 = m_2$).

Proof. Define $\mathrm{mmt}\to\mathrm{mt} := \mathrm{mApply}\ (_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*)) : M(M(T)) \to M(T)$. Since $T = \{\mathrm{tt}^*\}$ has a unique inhabitant, $\mathrm{mPure}_T = (_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*))$ by function extensionality, and for every $t \in T$ the left unit law (9) gives $\mathrm{mmt}\to\mathrm{mt}\big((\mathrm{mPure}_{M(T)} \circ \mathrm{mPure}_T)(t)\big) = \mathrm{mPure}_T(\mathrm{tt}^*)$. Applying the right unit law (10), associativity (11), the purity law (14), and the left unit law (9) yields:

$$\begin{aligned}
mt &= \mathrm{mApply}\ \mathrm{mPure}_T\ mt && \text{(by (10))}\\
&= \mathrm{mApply}\ (_ \mapsto \mathrm{mPure}_T(\mathrm{tt}^*))\ mt && (T = \{\mathrm{tt}^*\})\\
&= \mathrm{mApply}\ (t \mapsto \mathrm{mmt}\to\mathrm{mt}((\mathrm{mPure}_{M(T)} \circ \mathrm{mPure}_T)(t)))\ mt && \text{(by (9))}\\
&= \mathrm{mmt}\to\mathrm{mt}\big(\mathrm{mApply}\ (\mathrm{mPure}_{M(T)} \circ \mathrm{mPure}_T)\ mt\big) && \text{(by (11))}\\
&= \mathrm{mmt}\to\mathrm{mt}\big(\mathrm{mPure}_{M(T)}(mt)\big) && \text{(by (14))}\\
&= \mathrm{mPure}_T(\mathrm{tt}^*) && \text{(by (9)).}
\end{aligned}$$

□

5 Classification of Concrete Computational Effects

We now evaluate $\mathrm{IsPure}(M)$ across seven standard computational effects, giving full equational derivations for each case and summarizing the classification in Table 1.

Theorem 5.1. *Let* $\mathbf{SET}_\ell$ *be a universe of sets, and for a set* $A \in \mathrm{ob}(\mathbf{SET}_\ell)$ *write* $\mathrm{Subsingleton}(A) := (\forall a_1, a_2 \in A,\ a_1 = a_2)$.

(1) ***Partiality*** *(*Option***, with*** $\mathrm{mPure}(z) = \mathrm{some}(z)$ ***and*** $\mathrm{mApply}\ f\ \mathrm{none} = \mathrm{none}$***,*** $\mathrm{mApply}\ f\ (\mathrm{some}(z)) = f(z)$***):***

$$\neg\,\mathrm{IsPure}(\mathrm{Option}).$$

Proof: *Choose the witness computation* $mz = \mathrm{none} \in \mathrm{Option}(T)$. *Then the left-hand side and right-hand side of the purity equation* (14) *evaluate to:*

$$\mathrm{mApply}\ (\mathrm{mPure} \circ \mathrm{mPure})\ \mathrm{none} \;=\; \mathrm{none}, \qquad \mathrm{mPure}(\mathrm{none}) \;=\; \mathrm{some}(\mathrm{none}).$$

Thus (14) *forces* $\mathrm{none} = \mathrm{some}(\mathrm{none})$ *in* $\mathrm{Option}(\mathrm{Option}(T))$*, which is impossible since* none *and* some *are disjoint constructors.*

(2) ***Nondeterminism*** *(*List***, with*** $\mathrm{mPure}(z) = [z]$ ***and*** $\mathrm{mApply}\ f\ zs = \mathrm{concatMap}\ f\ zs$***):***

$$\neg\,\mathrm{IsPure}(\mathrm{List}).$$

Proof: *Choose the two-element list* $mz = [\mathrm{tt}^*, \mathrm{tt}^*] \in \mathrm{List}(T)$. *Since* $(\mathrm{mPure} \circ \mathrm{mPure})(z) = [[z]]$*, the left-hand side and right-hand side of* (14) *evaluate to:*

$$\mathrm{mApply}\ (\mathrm{mPure} \circ \mathrm{mPure})\ [\mathrm{tt}^*, \mathrm{tt}^*] \;=\; [[\mathrm{tt}^*], [\mathrm{tt}^*]], \qquad \mathrm{mPure}([\mathrm{tt}^*, \mathrm{tt}^*]) \;=\; [[\mathrm{tt}^*, \mathrm{tt}^*]].$$

Equating the two lists in $\mathrm{List}(\mathrm{List}(T))$ *and taking the tail (or comparing lengths* $2 \neq 1$*) yields* $[[\mathrm{tt}^*]] = []$*, a contradiction.*

(3) ***Environment reading*** *(*$\mathrm{Read}\ R$***, where*** $\mathrm{Read}\ R(Z) = (R \to Z)$***,*** $\mathrm{mPure}(z) = (_ \mapsto z)$***, and*** $(\mathrm{mApply}\ f\ m)(r) = f(m(r))(r)$***):***

$$\mathrm{IsPure}(\mathrm{Read}\ R) \implies \mathrm{Subsingleton}(R).$$

Proof: *Choose the identity reader* $mz = \mathrm{id}_R = (r \mapsto r) \in \mathrm{Read}\ R(R)$. *For any* $z \in R$, $(\mathrm{mPure} \circ \mathrm{mPure})(z) = (_ \mapsto (_ \mapsto z)) : R \to (R \to R)$. *Evaluating both sides of* (14) *on* $mz = \mathrm{id}_R$ *gives:*

$$\begin{aligned}
\mathrm{LHS} &= \mathrm{mApply}\ (\mathrm{mPure} \circ \mathrm{mPure})\ \mathrm{id}_R\\
&\quad = (r_1 \mapsto ((\mathrm{mPure} \circ \mathrm{mPure})(\mathrm{id}_R(r_1)))(r_1)) \;=\; (r_1 \mapsto (r_2 \mapsto r_1)),\\
\mathrm{RHS} &= \mathrm{mPure}(\mathrm{id}_R) \;=\; (r_1 \mapsto \mathrm{id}_R) \;=\; (r_1 \mapsto (r_2 \mapsto r_2)).
\end{aligned}$$

For any $r_1, r_2 \in R$, applying LHS = RHS *to r_1 and then to r_2 gives $r_1 = \text{LHS}(r_1)(r_2) = \text{RHS}(r_1)(r_2) = r_2$.*

(4) ***Logging (***Write W***, where*** Write $W(Z) = Z \times W$ ***for a monoid*** $(W, e, \bullet)$***,*** $\text{mPure}(z) = (z, e)$***, and*** $\text{mApply}\ f\ (z, w_0) = (y, w_0 \bullet w_1)$ ***where*** $(y, w_1) = f(z)$***):***

$$\text{IsPure}(\text{Write}\ W) \implies \text{Subsingleton}(W).$$

Proof: *Let $w \in W$ and consider the witness $mz = (e, w) \in \text{Write}\ W(W)$. Since* $(\text{mPure} \circ \text{mPure})(z) = ((z, e), e)$*, the two sides of* (14) *at $mz = (e, w)$ are:*

$$\begin{aligned} \text{LHS} &= \text{mApply}\ (\text{mPure} \circ \text{mPure})\ (e, w) \ =\ ((e, e),\ w \bullet e), \\ \text{RHS} &= \text{mPure}((e, w)) \ =\ ((e, w),\ e). \end{aligned}$$

Taking the first projection π_1 of LHS = RHS *gives $(e, e) = (e, w)$, and taking the second projection π_2 gives $e = w$. Hence for any $w_1, w_2 \in W$, $w_1 = e = w_2$.*

(5) ***Mutable state (***State S***, where*** State $S(Z) = (S \to Z \times S)$***,*** $\text{mPure}(z) = (s \mapsto (z, s))$***, and*** $(\text{mApply}\ f\ m)(s) = f(z)(s')$ ***where*** $(z, s') = m(s)$***):***

$$\text{IsPure}(\text{State}\ S) \implies \text{Subsingleton}(S).$$

Proof: *Choose the state-reading computation $mz = (s \mapsto (s, s)) \in \text{State}\ S(S)$. For any $z \in S$, we have*

$$(\text{mPure} \circ \text{mPure})(z) \ =\ \text{mPure}(s' \mapsto (z, s')) \ =\ \Big(s \mapsto ((s' \mapsto (z, s')),\ s)\Big).$$

Therefore, the left-hand side of (14) *applied to $mz = (s \mapsto (s, s))$ is:*

$$\begin{aligned} \text{LHS}(s_1) &= (\text{mApply}\ (\text{mPure} \circ \text{mPure})\ mz)(s_1) \\ &= ((\text{mPure} \circ \text{mPure})(s_1))(s_1) \ =\ ((s' \mapsto (s_1, s')),\ s_1), \end{aligned}$$

while the right-hand side of (14) *applied to $mz = (s \mapsto (s, s))$ is:*

$$\text{RHS}(s_1) \ =\ (\text{mPure}(mz))(s_1) \ =\ (mz,\ s_1) \ =\ ((s' \mapsto (s', s')),\ s_1).$$

For any $s_1, s_2 \in S$, evaluating LHS = RHS *at s_1 and taking the first projection π_1 yields the equality of functions $(s' \mapsto (s_1, s')) = (s' \mapsto (s', s'))$ in $S \to S \times S$. Evaluating both functions at $s' = s_2$ gives $(s_1, s_2) = (s_2, s_2)$, and taking the first projection π_1 yields $s_1 = s_2$.*

(6) ***Continuations (***Cont R***, where*** Cont $R(Z) = ((Z \to R) \to R)$***,*** $\text{mPure}(z) = (k \mapsto k(z))$***, and*** $(\text{mApply}\ f\ m)(k) = m(z \mapsto f(z)(k))$***):***

$$\text{IsPure}(\text{Cont}\ R) \implies \text{Subsingleton}(R).$$

Proof: *Let $r_1, r_2 \in R$, and consider the constant continuation computation $m_1 \coloneqq (_ \mapsto r_1) \in \text{Cont}\ R(R) = ((R \to R) \to R)$, which ignores its continuation argument $k : R \to R$ and returns r_1. In* Cont $R($Cont $R(R))$*, an outer continuation is a function $K :$ Cont $R(R) \to R$. The left-hand side of* (14) *at $mz = m_1$, applied to any outer continuation $K :$ Cont $R(R) \to R$, evaluates to:*

$$\begin{aligned} \text{LHS}(K) &= (\text{mApply}\ (\text{mPure} \circ \text{mPure})\ m_1)(K) \\ &= m_1(z \mapsto ((\text{mPure} \circ \text{mPure})(z))(K)) \ =\ r_1, \end{aligned}$$

because $m_1 = (_ \mapsto r_1)$ returns r_1 regardless of its argument. Conversely, the right-hand side of (14) *at $mz = m_1$, applied to K, evaluates to:*

$$\text{RHS}(K) \ =\ (\text{mPure}(m_1))(K) \ =\ K(m_1).$$

Now choose the constant outer continuation $K_2 \coloneqq (_ \mapsto r_2) : \mathrm{Cont}\ R(R) \to R$. *Evaluating* LHS = RHS *at* $K = K_2$ *gives*

$$r_1 \;=\; \mathrm{LHS}(K_2) \;=\; \mathrm{RHS}(K_2) \;=\; K_2(m_1) \;=\; r_2.$$

(7) ***Input/Output and Exceptions (***$\mathrm{ErrorState}\ E\ S$***, where*** $\mathrm{ErrorState}\ E\ S(Z) = (S \to (E \times S) \uplus (Z \times S))$***,*** $\mathrm{mPure}(z) = (s \mapsto \mathrm{inr}(z, s))$***, and*** $\mathrm{mApply}\ f\ m$ ***propagates*** $\mathrm{inl}(e, s')$ ***and applies*** $f(z)(s')$ ***on*** $\mathrm{inr}(z, s')$***):***

$$\forall e \in E,\ \forall s \in S, \qquad \neg\,\mathrm{IsPure}(\mathrm{ErrorState}\ E\ S).$$

Proof: *Given* $e \in E$ *and* $s \in S$*, define the failing computation* $m \coloneqq (s' \mapsto \mathrm{inl}(e, s')) \in \mathrm{ErrorState}\ E\ S(T)$. *Evaluating both sides of* (14) *on* m *and then at initial state* $s \in S$ *yields:*

$$(\mathrm{mApply}\ (\mathrm{mPure} \circ \mathrm{mPure})\ m)(s) \;=\; \mathrm{inl}(e, s), \qquad (\mathrm{mPure}(m))(s) \;=\; \mathrm{inr}(m, s),$$

contradicting the disjointness of inl *and* inr.

Monad M	**Carrier** $M(Z)$	**Terminal** $M(T)$	**Witness** mz	**LHS vs. RHS of** (14)	$\mathrm{IsPure}(M)$
Option	$1 + Z$	$1 + 1$	none	$\mathrm{none} \neq \mathrm{some}(\mathrm{none})$	False
List	$\mathrm{List}(Z)$	$\mathrm{List}(1) \cong \mathbb{N}$	$[\mathrm{tt}^*, \mathrm{tt}^*]$	$[[\mathrm{tt}^*], [\mathrm{tt}^*]] \neq [[\mathrm{tt}^*, \mathrm{tt}^*]]$	False
Read R	$R \to Z$	$R \to 1 \cong 1$	id_R	$(r_1, r_2 \mapsto r_1) = (r_1, r_2 \mapsto r_2)$	$\iff \lvert R\rvert \leq 1$
Write W	$Z \times W$	$1 \times W \cong W$	(e, w)	$((e, e), w) = ((e, w), e)$	$\iff \lvert W\rvert \leq 1$
State S	$S \to Z \times S$	$S \to 1 \times S \cong S^S$	$s \mapsto (s, s)$	$((s' \mapsto (s_1, s')), s_1) = (mz, s_1)$	$\iff \lvert S\rvert \leq 1$
Cont R	$(Z \to R) \to R$	$(1 \to R) \to R \cong R^R$	$_ \mapsto r_1$	$(_ \mapsto r_1) = (K \mapsto K(m_1))$	$\iff \lvert R\rvert \leq 1$
$\mathrm{ErrorState}\ E\ S$	$S \to (E + Z) \times S$	$S \to (E + 1) \times S$	$s' \mapsto \mathrm{inl}(e, s')$	$\mathrm{inl}(e, s) \neq \mathrm{inr}(m, s)$	False $(E, S \neq \emptyset)$

Table 1: Summary of the seven standard computational effects, their terminal carriers $M(T)$, their purity witness computations, and their exact $\mathrm{IsPure}(M)$ classification (where $\lvert A\rvert \leq 1$ denotes $\mathrm{Subsingleton}(A)$).

5.1 Strict Separation Between Basic Functional Category Purity and Terminal Subsingleton Triviality

As Table 1 illustrates, the terminal subsingleton condition $\mathrm{MonadicTerminalTypeSubSingleton}(M)$ ($M(T) \cong T$, Theorem 4.4) suffices to rule out Option, List, Write W ($\lvert W\rvert > 1$), State S ($\lvert S\rvert > 1$), Cont R ($\lvert R\rvert > 1$), and $\mathrm{ErrorState}\ E\ S$ ($E, S \neq \emptyset$). However, Read R satisfies

$$\mathrm{Read}\ R(T) \;=\; (R \to T) \;\cong\; T$$

for *every* environment set R. We now formulate and prove a formal separation theorem showing that the Basic Functional Category condition on $(\mathbf{K}(M), \mathrm{KFF})$ (and in particular the existence of a natural transformation $\eta : \mathrm{Id}_{\mathbf{K}(M)} \Rightarrow G$ satisfying the terminal unit law (4)) is **strictly stronger** than the terminal subsingleton property $\mathrm{MonadicTerminalTypeSubSingleton}(M)$.

Theorem 5.2 (Strict separation theorem)**.** *For monads* M *on* $\mathbf{SET}_\ell$*:*

(i) ***Universal Implication:*** *For every monad* M *on* $\mathbf{SET}_\ell$*,*

$$\mathrm{BasicFunctionalCategory}(\mathbf{K}(M), \mathrm{KFF}) \implies \mathrm{MonadicTerminalTypeSubSingleton}(M).$$

(ii) ***Strictness (Failure of the Converse):*** *For every set* $R \in \mathrm{ob}(\mathbf{SET}_\ell)$ *containing at least two distinct elements* $r_1 \neq r_2$ *(for instance,* $R = \{0, 1\}$*), the reader monad* $M = \mathrm{Read}\ R$ *satisfies:*

(a) MonadicTerminalTypeSubSingleton(Read R) ***holds*** *($\forall mt \in$ Read $R(T)$, $mt = \mathrm{mPure}_T(\mathrm{tt}^*)$);*

(b) BasicFunctionalCategory($\mathbf{K}$(Read R), KFF) ***fails*** *(no natural transformation $\eta : \mathrm{Id}_{\mathbf{K}(\mathrm{Read}\ R)} \Rightarrow G$ satisfies the terminal unit law (4)).*

Proof. **Part (i):** Suppose $(\mathbf{K}(M), \mathrm{KFF})$ is a Basic Functional Category. By Theorem 4.3, M satisfies Idempotent(M), which by Lemma 4.2 is equivalent to IsPure(M). Applying Theorem 4.4 yields MonadicTerminalTypeSubSingleton(M).

Part (ii)(a): Let $mt \in$ Read $R(T) = (R \to T)$. For every $r \in R$, since $T = \{\mathrm{tt}^*\}$ has a unique inhabitant, $mt(r) = \mathrm{tt}^* = (_ \mapsto \mathrm{tt}^*)(r) = \mathrm{mPure}_T(\mathrm{tt}^*)(r)$. By function extensionality, $mt = \mathrm{mPure}_T(\mathrm{tt}^*)$, so MonadicTerminalTypeSubSingleton(Read R) holds for every set R.

Part (ii)(b): Suppose for contradiction that $(\mathbf{K}(\mathrm{Read}\ R), \mathrm{KFF})$ is a Basic Functional Category (or merely that there exists a natural transformation $\eta : \mathrm{Id}_{\mathbf{K}(\mathrm{Read}\ R)} \Rightarrow G$ satisfying the terminal unit law (4)). By Theorem 4.3(i) and Lemma 4.2, IsPure(Read R) must hold. By Theorem 5.1(3), IsPure(Read R) implies Subsingleton(R), forcing $r_1 = r_2$, which contradicts $r_1 \neq r_2$.

To see this failure directly on the explicit separating model $M =$ Read R with two distinct environment values $r_1 \neq r_2 \in R$ (such as the two-element boolean environment $R = \{0, 1\}$ with $r_1 = 0$ and $r_2 = 1$), note that in $\mathbf{K}$(Read R), the candidate unit forced by (3) is $\eta_Z(z) = (r_{\mathrm{outer}} \mapsto (_ \mapsto (r_{\mathrm{inner}} \mapsto z)))$, and at $Z = T$ the terminal law (4) forces $\eta_T(\mathrm{tt}^*) = (r_{\mathrm{outer}} \mapsto (_ \mapsto (r_{\mathrm{inner}} \mapsto \mathrm{tt}^*)))$. Consider the environment-reading Kleisli morphism

$$f \;:=\; (_ \mapsto \mathrm{id}_R) \;=\; \big(t \mapsto (r \mapsto r)\big) \;\in\; \mathrm{Hom}_{\mathbf{K}(\mathrm{Read}\ R)}(T, R).$$

Evaluating the naturality equation $f \star_{\mathbf{K}} \eta_R = \eta_T \star_{\mathbf{K}} G(f)$ at $\mathrm{tt}^* \in T$, outer environment $r_1 \in R$, dummy terminal argument $\mathrm{tt}^* \in T$, and inner environment $r_2 \in R$ gives:

$$\begin{aligned}
((f \star_{\mathbf{K}} \eta_R)(\mathrm{tt}^*)(r_1)(\mathrm{tt}^*))(r_2) &= (\eta_R(f(\mathrm{tt}^*)(r_1))(r_1)(\mathrm{tt}^*))(r_2) \;=\; (\eta_R(r_1)(r_1)(\mathrm{tt}^*))(r_2) \;=\; r_1, \\
((\eta_T \star_{\mathbf{K}} G(f))(\mathrm{tt}^*)(r_1)(\mathrm{tt}^*))(r_2) &= (G(f)(\eta_T(\mathrm{tt}^*)(r_1))(r_1)(\mathrm{tt}^*))(r_2) \\
&= f(\eta_T(\mathrm{tt}^*)(r_1)(\mathrm{tt}^*)(r_2))(r_2) \;=\; f(\mathrm{tt}^*)(r_2) \;=\; r_2.
\end{aligned}$$

On the left-hand side, binding f before η_R reads the *outer* environment r_1 and freezes it inside the thunk; on the right-hand side, postcomposing f inside $G(f)$ defers reading the environment until the *inner* evaluation at r_2. Since $r_1 \neq r_2$, naturality of η fails for Read R whenever $|R| \geq 2$, even though Read $R(T) = (R \to T) \cong T$ is a subsingleton. □

5.2 Scope, Implications for Commutative Effects, and Limitations of the Framework

Our results establish a sharp boundary for the pointfree Basic Functional Category and CoEndoYoneda framework, which clarifies both its power as a purity specification and its scope for modeling effects:

(1) **Exclusion of non-idempotent and commutative effects:** By Theorem 4.3, for a Kleisli category $\mathbf{K}(M)$ equipped with the canonical identity-on-objects embedding KFF : $\mathbf{SET}_\ell \to \mathbf{K}(M)$, the Basic Functional Category structure (and hence the CoEndoYoneda Isomorphism Theorem 3.4) holds if and only if M is an idempotent monad. Consequently, the CoEndoYoneda isomorphism with respect to KFF does *not* apply to Kleisli categories of non-idempotent monads such as Option, List, State S ($|S| \geq 2$), Cont R ($|R| \geq 2$), or ErrorState E S ($E, S \neq \emptyset$). Moreover, unlike premonoidal centrality [15, 9]—which requires effect commutativity $(f \ltimes h) = (f \rtimes h)$ and therefore includes commutative effects

such as Read R and commutative Write W in the center—our condition requires that *every* computation commute with the thunking endofunctor $G(Z) = (T \to M(Z))$ via η. As shown in Theorem 5.2, for Read R ($|R| \geq 2$), binding an environment query before η freezes the outer environment r_1, whereas pushing the query into G reads the inner environment r_2; for Write W ($|W| \geq 2$), binding a log emission before η emits the log at the outer level, whereas pushing it into G traps the log inside the thunk $T \to (Z \times W)$. Thus, the Basic Functional Category axiom characterizes *strict effectlessness* (referential transparency across thunking) rather than effect commutativity.

(2) **Which non-trivial monads are idempotent?** Although the seven effect families in Theorem 5.1 collapse to trivial parameter sets when idempotent, idempotent monads in general category theory are far from trivial: on a general category $\mathcal{C}$, idempotent monads classify *reflective subcategories* (and localization / modality operators, such as open/closed modalities, sheafification, or propositional truncation in homotopy type theory) [1, 7, 11]. On $\mathbf{SET}_\ell$, where every non-empty set splits over T, the only reflective subcategories are the full category $\mathbf{SET}_\ell$ ($M \cong \mathrm{Id}_{\mathbf{SET}_\ell}$) and the terminal subcategory ($M(Z) \cong T$, or $M(\emptyset) = \emptyset$ and $M(Z) \cong T$ for $Z \neq \emptyset$). Thus, on $\mathbf{SET}_\ell$, our characterization accurately certifies that a monadic program category $(\mathbf{K}(M), \mathrm{KFF})$ behaves purely if and only if M introduces no computational side effects beyond pure values (or trivial termination).

(3) **Alternative pointfree characterizations for benign or commutative effects:** If one wishes to relax strict purity to admit specific classes of benign effects in a pointfree setting, our hierarchy of theorems suggests three natural directions:

(a) *Discardable / read-only effects* (such as Read R): characterized by the strictly weaker terminal subsingleton law $M(T) \cong T$ (Theorem 4.4), which demands that computations returning no data carry no observable side-effect state, and corresponds categorically to the terminal object $\mathrm{KFF}(T)$ remaining terminal in $\mathbf{K}(M)$ (semicartesian / affine structure);

(b) *Commutative effects* (such as Read R and commutative Write W): characterized pointfreely by central morphisms in premonoidal and Freyd categories [15, 9];

(c) *Restricted or relative CoEndoYoneda structures:* restricting the naturality of $\eta : \mathrm{Id}_{\mathcal{C}} \Rightarrow G$ to the subcategory of pure (or thunkable [6]) morphisms inside an ambient effectful category $\mathcal{C}$, or varying the functional functor $\mathrm{FF} : \mathbf{SET}_\ell \to \mathcal{C}$ beyond the identity-on-objects Kleisli inclusion KFF.

6 Formal Verification in Cubical Agda and Lean 4

Both formalizations are 100% machine-checked with zero postulates, zero unproved axioms, and zero `sorry`s. Table 2 provides the exact alignment between the mathematical results in this paper, the Cubical Agda development (`CoEndoYoneda.lagda.md` [3]), and the Palomar-verified Lean 4 formalization (`coEndoYoneda-and-purity` [4]).

References

[1] F. Borceux, *Handbook of Categorical Algebra 2: Categories and Structures*, Encyclopedia of Mathematics and its Applications, vol. 51, Cambridge University Press, 1994.

[2] L. Duponcheel, *A pointfree Yoneda lemma for endofunctors of functional categories*, Working paper, https://www.academia.edu/109957588.

[3] L. Duponcheel, *Cubical Agda formalization: CoEndoYoneda and its relation to Purity*, https://github.com/LucDuponcheelAtGitHub/AgdaCoEndoYonedaAndPurity, 2026.

Paper Reference	Cubical Agda (`CoEndoYoneda.lagda.md`)	Lean 4 (`CoEndoYonedaAndPurity`)	Status
Definition 2.4	`BasicFunctionalCategory`	`BasicFunctionalCategory`	Verified
Lemma 2.3	`lemma`, `lemma-T`	`lemma_g`, `lemma_T`	Proved
Proposition 2.5	η`-eq-T`	η`_eq_T`	Proved
Definition 3.1	`FunctionalCategory`	`FunctionalCategory`	Verified
Theorem 3.4(i)	`fz→τz_commute`	`fz_to_τz_commute`	Proved
Theorem 3.4(ii)	`to_from_proof`	`to_from_proof_hom`, `to_from_proof`	Proved
Theorem 3.4(iii)	`from_to_proof'`, `from_to_proof`	`from_to_proof'`, `from_to_proof`	Proved
Theorem 3.4 (Iso)	`coEndoYonedaIso`	`coEndoYonedaIso`	Proved
Proposition 3.5	η`FFT-is-`η`-at-FFT`	η`FFT_is_`η`_at_FFT`	Proved
Proposition 3.6	`SetFunctionalCategory`	`set_is_functional_category`	Proved
Lemma 4.2	`idempotent-equiv-is-pure`	`idempotent_iff_isPure`	Proved
Theorem 4.3	`idempotent-equiv-kleisli-...`	`kleisliBasicFunctionalCategory_iff_...`	Proved
Theorem 4.4	`is-pure-implies-monadic-...`	`isPure_implies_monadicTerminalType...`	Proved
Theorem 5.1(1)–(2)	`maybe-not-pure`, `list-not-pure`	`maybe_not_pure`, `list_not_pure`	Proved
Theorem 5.1(3)–(6)	`read/write/state/cont-purity-implies`	`read/write/state/cont_purity_implies`	Proved
Theorem 5.1(7)	`io-not-pure`	`io_not_pure`	Proved

Table 2: Alignment between the paper, the Cubical Agda formalization, and the Lean 4 Palomar formalization.

[4] L. Duponcheel, *Palomar Lean 4 formalization: CoEndoYoneda and Purity*, https://github.com/LucDuponcheelAtGitHub/coEndoYoneda-and-purity, 2026.

[5] L. Duponcheel, *Program Specification Based Programming*, https://github.com/LucDuponcheelAtGitHub/AgdaPSBP, 2026.

[6] C. Führmann, *Direct models of the computational lambda-calculus*, Electronic Notes in Theoretical Computer Science **20** (1999), 245–292, https://doi.org/10.1016/S1571-0661(04)80078-1.

[7] G. M. Kelly, *A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on*, Bulletin of the Australian Mathematical Society **22** (1980), no. 1, 1–83, https://doi.org/10.1017/S0004972700006353.

[8] F. W. Lawvere, *Functorial semantics of algebraic theories*, Proceedings of the National Academy of Sciences **50** (1963), no. 5, 869–872, https://doi.org/10.1073/pnas.50.5.869.

[9] P. B. Levy, J. Power, and H. Thielecke, *Modelling environments in call-by-value programming languages*, Information and Computation **185** (2003), no. 2, 182–210, https://doi.org/10.1016/S0890-5401(03)00088-9.

[10] E. Moggi, *Notions of computation and monads*, Information and Computation **93** (1991), no. 1, 55–92, https://doi.org/10.1016/0890-5401(91)90052-4.

[11] nLab authors, *Idempotent monad*, https://ncatlab.org/nlab/show/idempotent+monad.

[12] nLab authors, *Well-pointed endofunctor*, https://ncatlab.org/nlab/show/well-pointed+endofunctor.

[13] G. Plotkin and J. Power, *Notions of computation determine monads*, in: *Foundations of Software Science and Computation Structures (FoSSaCS 2002)*, Lecture Notes in Computer Science, vol. 2303, Springer, 2002, pp. 342–356, https://doi.org/10.1007/3-540-45931-6_24.

[14] G. Plotkin and J. Power, *Algebraic operations and generic effects*, Applied Categorical Structures **11** (2003), no. 1, 69–94, https://doi.org/10.1023/A:1023064908962.

[15] J. Power and E. Robinson, *Premonoidal categories and notions of computation*, Mathematical Structures in Computer Science **7** (1997), no. 5, 453–468, https://doi.org/10.1017/S0960129597002375.

[16] J. Power and H. Thielecke, *Closed Freyd- and κ-categories*, in: *Automata, Languages and Programming (ICALP 1999)*, Lecture Notes in Computer Science, vol. 1644, Springer, 1999, pp. 625–634, https://doi.org/10.1007/3-540-48523-6_59.