Presheaves as an Purified Effectful Call-by-Value Language

Pierre-Marie Pédrot

September 5, 2019

Abstract

This note shows that the well-known presheaf construction can be understood as a standard realizability technique applied to an effectful language. More specifically, the *naturality* requirement of presheaf morphisms is tantamount to a semantic property of purity, and the category of presheaves is thus the restriction to observationally pure terms of an effectful call-by-value language.

1 Categorical Nonsense

For the sake of completeness, we recall here a few categorically nonsensical definitions and results.

1.1 Presheaves

Definition 1. Let $\mathbf{C}$ be some category, henceforth called the *base category.* A presheaf over $\mathbf{C}$ is just a functor $\mathbf{C}^{op} \rightarrow \mathbf{Set}$.

Theorem 1. *Presheaves with natural transformations as morphisms form a cocomplete cartesian closed category.*

This category, written thereafter $\hat{\mathbf{C}}$, is a actually much richer, e.g. it is also a *topos*, but we will not care about this in this note. Let us also remark that this definition is incredibly compact, which is probably one of the reasons of its ubiquity in categorical models.

Insight 1. Presheaf categories are the bread and butter of model constructions, a well-known fact better summarized as: "*Préfaisceaux: retour de l'être aimé, désenvoûtement, démarrage de motos russes*"[1].

[1] I am sure Grothendieck would have approved this statement.

1.2 Thunkability

Definition 2. Let $\mathbf{L}$ be a category equipped with a monad (T, η, μ). We say that a morphism $f : \mathbf{L}(A, T\,B)$ is *thunkable* [Füh99] whenever the following diagram commutes.

$$A \xrightarrow{f} T\,B \underset{\eta_{T\,B}}{\overset{T\,\eta_B}{\rightrightarrows}} T^2\,B$$

Thunkability captures the intuitive notion of being observationally pure in call-by-value.

Insight 2. In call-by-value λ-calculus, a term $t : A$ is thunkable if it satisfies the equation below.

$$\texttt{let}\ x\ := t\ \texttt{in}\ \lambda().\,x \equiv \lambda().\,t.$$

This intuitively reflects the fact that t behaves as a value, as evaluating it eagerly doesn't have any observable difference from thunking it.

Proposition 1. *We have the following.*

- *Every morphism of the form* $A \xrightarrow{f} B \xrightarrow{\eta_B} T\,B$ *is thunkable.*
- *In particular,* η_A *is thunkable.*
- *If* $f : \mathbf{L}(A, T\,B)$ *and* $g : \mathbf{L}(B, T\,C)$ *are both thunkable, then so is*

$$A \xrightarrow{f} T\,B \xrightarrow{T\,g} T^2\,B \xrightarrow{\mu_B} T\,B.$$

This immediately leads to the correctness of the following definition.

Definition 3. The Führmann category $\mathbf{L}_{\Theta(T)}$ is the subcategory of the Kleisli category $\mathbf{L}_T$ restricted to thunkable morphisms.

Proposition 2. *Assuming* $\mathbf{L}$ *has equalizers and* T *is a strong monad, if* $\mathbf{L}$ *is cartesian closed then* $\mathbf{L}_{\Theta(T)}$ *is also cartesian closed. Note that in general the whole Kleisli category* $\mathbf{L}_T$ *is* not *cartesian closed.*

Proof. This is a folklore result from realizability. Define:

- $1_{\mathbf{L}_{\Theta(T)}} := 1_{\mathbf{L}}$
- $A \times_{\mathbf{L}_{\Theta(T)}} B := A \times_{\mathbf{L}} B$
- $A \Rightarrow_{\mathbf{L}_{\Theta(T)}} B$ is intuitively $\{f : A \Rightarrow_{\mathbf{L}} T\,B \mid \forall x : A.\, f\,x \text{ is thunkable}\}$. Formally, we define it as an equalizer in $\mathbf{L}$:

$$(A \Rightarrow_{\mathbf{L}_{\Theta(T)}} B) \longrightarrow (A \Rightarrow_{\mathbf{L}} T\,B) \underset{A \Rightarrow \eta_{T\,B}}{\overset{A \Rightarrow T\,\eta_B}{\rightrightarrows}} (A \Rightarrow_{\mathbf{L}} T^2\,B).$$

The corresponding morphisms are straightforward, but the fact that they enjoy their respective universal property is critically due to the thunkability restriction. □

Insight 3. This construction is pervasive in realizability when the language of realizers is call-by-value. For instance, when the only effect is divergence, a term is thunkable iff it has a weak normal form. Compare with the standard way to define realizers for implication in Kleene realizability [Kle45]:

$$f \Vdash A \to B \qquad \equiv \qquad \forall x.\, x \Vdash A \to f\ x \downarrow v \wedge v \Vdash B.$$

This is hardwiring the fact that we only consider functions that, when applied to a well-behaved argument, result in a thunkable term. As such, this is literally the thunkable arrow from the definition above. This construction can be readily generalized to any effectful language of realizers by replacing $t \downarrow v$ with the adequate notion of being thunkable.

Barring logical consequences, it turns out that under mild assumptions, the thunkable restriction is essentially the identity [Lev17].

Definition 4. We say that T is of codescent type whenever the following diagram is an equalizer.

$$\begin{tikzcd} A \arrow[r, "\eta_A"] & T\,A \arrow[r, shift left, "T\,\eta_A"] \arrow[r, shift right, "\eta_{T\,A}"'] & T^2\,A \end{tikzcd}$$

In **Set**, almost all monads are of codescent type, except those isomorphic to the singleton monad or to the squash monad.

Proposition 3. *If T is of codescent type, then $\mathbf{L}_{\Theta(T)}$ is equivalent to $\mathbf{L}$.*

Proof. In that case, $f : \mathbf{L}(A, T\,B)$ is thunkable iff there exists $\hat{f} : \mathbf{L}(A, B)$ s.t. $f = \eta_B \circ \hat{f}$. □

Insight 4. The above result is essentially saying that thunkable terms are extensionally equivalent to some value in the model. It might not be the case in the source syntax though.

1.3 Call-by-Value Effects in Direct Style

This section presents a variant of the previous definitions, although they are describing this time call-by-value effects in direct style. We stick to Führmann presentation for simplicity, although we could also have chosen a finer-grained model like CBPV [Lev04].

Definition 5. Let $F, G : \mathbf{C} \to \mathbf{D}$ be two functors. An artificial transformation $\theta : F \overset{♻}{\to} G$ is a transformation that is not necessarily natural, that is, it is a family of morphisms $\theta_A : \mathbf{D}(F\,A, G\,A)$ not expected to satisfy any further equation.

Definition 6. An abstract Kleisli category is given by:

- A category $\mathbf{K}$

- A functor $L : \mathbf{K} \to \mathbf{K}$
- An *artificial* transformation $\vartheta : 1 \overset{♻}{\to} L$
- A *natural* transformation $\varepsilon : L \to 1$

such that ϑ_L is natural, and subject to the following equations.

$$
\begin{tikzcd}
1 \arrow[r, "\vartheta"] \arrow[d, "\vartheta"'] & L \arrow[d, "\vartheta_L"] \\
L \arrow[r, "L\,\vartheta"] & L^2
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[r, "\vartheta"] \arrow[dr, "="'] & L \arrow[d, "\varepsilon"] \\
 & 1
\end{tikzcd}
\qquad
\begin{tikzcd}
L \arrow[r, "\vartheta_L"] \arrow[dr, "="'] & L^2 \arrow[d, "L\,\varepsilon"] \\
 & L
\end{tikzcd}
$$

Insight 5. This is categorical gobbledygook capturing a very weak direct style notion call-by-value, where $L\,A$ should be understood as the type of thunks returning A. Nothing that looks like a proper type former in sight, and not even a syntactic notion of value. The very fact we do not require ϑ to be natural is categorically smelly, and likely a manifestation of the deeply entrenched bias of categorical semantics for the negative fragment and call-by-name in general.

Proposition 4. *If* $\mathbf{L}$ *is a category equipped with a monad* (T, η, μ)*, then the Kleisli category* $\mathbf{L}_T$ *is also an abstract Kleisli category, with* $L\,A := T\,A$ *on objects, for any* $f : \mathbf{L}_T(A, B)$*,* $L\,f := \eta \circ \mu \circ T\,f$*,* $\vartheta := \eta_T \circ \eta$ *and* $\varepsilon := \mathtt{id}$*.*

$\mathbf{K}$ stands for a call-by-value language where effects are ambient. Thunkability is readily generalized to this setting.

Definition 7. A morphism $f : \mathbf{K}(A, B)$ is thunkable whenever the following diagram commutes.

$$
\begin{tikzcd}
A \arrow[r, "\vartheta_A"] \arrow[d, "f"'] & L\,A \arrow[d, "L\,f"] \\
B \arrow[r, "\vartheta_B"] & L\,B
\end{tikzcd}
$$

Note that on an actual Kleisli category, this definition coincides with the one given in that setting.

2 Artificial Presheaves

For the remainder of this section, let us pick $\mathbf{C}$ a base category.

Definition 8. The category of artificial presheaves $\mathbf{C}^{♻}$ is defined as follows.

- Its objects are functors $\mathbf{C}^{op} \to \mathbf{Set}$.
- $\mathbf{C}^{♻}(A, B)$ are artificial transformations $A \overset{♻}{\to} B$.

The goal of this section is to show the following theorem.

Theorem 2. $\mathbf{C}^{♻}$ *is an abstract Kleisli category.*

2.1 Notations

Bear in mind that we are dealing with higher-order constructions, so that it is easy to get lost trying to remember what level we are currently living at. For readability, we use a fair amount of notations.

We will use type-theoretic notations for the **Set** category. **Set** happens to be a model of Martin-Löf type-theory, so there is no reason not to take advantage of this fact. In particular given a set A and an A-indexed set B, $\Pi(x : A).\, B$ stands for the set-theoretic dependent function space, which should not be confused with $\forall(x : A).\, B$. While the former is the set, the latter is a first-order formula.

Letters p, q, r will range over objects of $\mathbf{C}$ and greek letters α, β, γ will range over morphisms of $\mathbf{C}$. Given $p : \mathbf{C}$, we write $(q\,\alpha : p)$ for binders $(q : \mathbf{C})\,(\alpha : \mathbf{C}(q, p))$. For instance,

$$\forall(q\,\alpha : p).\, A \quad \equiv \quad \forall(q : \mathbf{C})\,(\alpha : \mathbf{C}(q, p)).\, A.$$

To disambiguate them from higher-order functors defined later on, functors $A : \mathbf{C}^{\text{✿}} \equiv \mathbf{C}^{op} \to \mathbf{Set}$ will be called presheaves, and will be given by a pair (A_p, θ_A) where A is a family of sets indexed by objects of $\mathbf{C}$ and

$$\theta_A : \Pi(p : \mathbf{C})\,(q\,\alpha : p).\, A_p \to A_q$$

subject to the usual equations for functors, i.e.

$$\theta_A\ \mathtt{id}_p^{\mathbf{C}} = \mathtt{id}_{A_p}^{\mathbf{Set}} \qquad\qquad \theta_A\ (\beta \circ^{\mathbf{C}} \alpha) = \theta_A\ \beta \circ^{\mathbf{Set}} \theta_A\ \alpha.$$

As already done in the above equations, we will omit the p and q arguments of θ_A.

2.2 Defining L

We need to define $L : \mathbf{C}^{\text{✿}} \to \mathbf{C}^{\text{✿}}$, both on objects and morphisms of $\mathbf{C}^{\text{✿}}$.

Object component The object component is a presheaf, so it is defined as a pair. Let $(A, \theta_A) : \mathbf{C}^{\text{✿}}$, we build $(L\,A, \theta_{L\,A}) : \mathbf{C}^{\text{✿}}$ as follows.

$$\begin{array}{rcl} (L\,A)_p & : & \mathbf{Set} \\ (L\,A)_p & := & \Pi(q\,\alpha : p).\, A_q \end{array}$$

Given $\alpha : \mathbf{C}(q, p)$ we define

$$\begin{array}{rcl} \theta_{L\,A}\ \alpha & : & (L\,A)_p \to (L\,A)_q \\ \theta_{L\,A}\ \alpha & := & \lambda(x : \Pi(r\,\beta : p).\, A_r)\,(r\,\beta : q).\, x\ r\ (\beta \circ \alpha) \end{array}$$

It is easy to check that indeed $(L\,A, \theta_{L\,A})$ is functorial, owing to the fact that $\mathbf{C}$ itself is a category, as

$$\begin{aligned} \theta_{L\,A}\ \mathtt{id}_p^{\mathbf{C}} &= \lambda(x : \Pi(r\,\beta : p).\, A_r)\,(r\,\beta : p).\, x\ r\ (\beta \circ \mathtt{id}_p^{\mathbf{C}}) \\ &= \lambda(x : \Pi(r\,\beta : p).\, A_r)\,(r\,\beta : p).\, x\ r\ \beta \\ &= \mathtt{id}_p^{\mathbf{Set}} \end{aligned}$$

and similarly for $\alpha : \mathbf{C}(q, p)$ and $\beta : \mathbf{C}(r, q)$ we have

$$\begin{aligned}
\theta_{L\,A}\ (\beta \circ \alpha) &= \lambda(x : \Pi(s\,\gamma : p).\, A_r)\,(s\,\gamma : p).\, x\ r\ (\gamma \circ (\beta \circ \alpha)) \\
&= \lambda(x : \Pi(s\,\gamma : p).\, A_r)\,(s\,\gamma : p).\, x\ r\ ((\gamma \circ \beta) \circ \alpha) \\
&= (\theta_{L\,A}\ \beta) \circ^{\mathbf{Set}} (\theta_{L\,A}\ \alpha).
\end{aligned}$$

Morphism component Let (A_p, θ_A) and (B_p, θ_B) be two presheaves and $f : \mathbf{C}^{✿}(A, B)$. Recall that

$$\mathbf{C}^{✿}(A, B) \equiv \Pi(p : \mathbf{C}).\, A_p \to B_p.$$

We define $L\,f : \mathbf{C}^{✿}(L\,A, L\,B)$ as follows.

$$\begin{array}{lcl}
(L\,f)_p & : & (L\,A)_p \to (L\,B)_p \\
(L\,f)_p & := & \lambda(x : \Pi(q\,\beta : p).\, A_q).\, \lambda(q\,\beta : p).\, f_q\ (x\ q\ \beta)
\end{array}$$

No naturality equation has to be checked, which is a good thing because there is none in general.

2.3 Defining ϑ

Let us now define $\vartheta : 1 \xrightarrow{✿} L$. Assume $(A_p, \theta_A) : \mathbf{C}^{✿}$, we need to define $\vartheta_A : \mathbf{C}^{✿}(A, L\,A)$.

$$\begin{array}{lcl}
(\vartheta_A)_p & : & A_p \to (L\,A)_p \\
(\vartheta_A)_p & := & \lambda(x : A_p).\, \lambda(q\,\alpha : p).\, \theta_A\ \alpha\ x
\end{array}$$

No naturality equations are expected.

2.4 Defining ε

Let us now define $\varepsilon : L \to 1$. Assume $(A_p, \theta_A) : \mathbf{C}^{✿}$, we need first to define $\varepsilon_A : \mathbf{C}^{✿}(L\,A, A)$.

$$\begin{array}{lcl}
(\varepsilon_A)_p & : & (L\,A)_p \to A_p \\
(\varepsilon_A)_p & := & \lambda(x : \Pi(q\,\alpha : p).\, A_q).\, x\ p\ \mathtt{id}_p
\end{array}$$

We are not done yet, as we need to check that ε is natural, that is, that the following diagram commutes for any $f : \mathbf{C}^{✿}(A, B)$.

$$\begin{tikzcd}
L\,A \arrow[r, "\varepsilon_A"] \arrow[d, "L\,f"'] & A \arrow[d, "f"] \\
L\,B \arrow[r, "\varepsilon_B"] & B
\end{tikzcd}$$

or, syntactically, this amounts to prove that for any $x : \Pi(q\,\alpha : p).\, A_q$ we have

$$\begin{aligned}
f_p\ ((\varepsilon_A)_p\ x) &\stackrel{?}{=} (\varepsilon_B)_p\ ((L\,f)_p\ x) \\
f_p\ (x\ p\ \mathtt{id}_p) &\stackrel{?}{=} (\lambda(q\,\alpha : p).\, f_q\ (x\ q\ \alpha))\ p\ \mathtt{id}_p \\
f_p\ (x\ p\ \mathtt{id}_p) &\stackrel{?}{=} f_p\ (x\ p\ \mathtt{id}_p).
\end{aligned}$$

2.5 Equations

We need to check a handful of equations.

Naturality of ϑ_L We show that $\vartheta_L : L \to L^2$ is natural, i.e. for any $f : \mathbf{C}^{\text{✿}}(A, B)$

$$\begin{tikzcd} L\,A \arrow[r, "\vartheta_{L\,A}"] \arrow[d, "L\,f"'] & L^2\,A \arrow[d, "L^2\,f"] \\ L\,B \arrow[r, "\vartheta_{L\,B}"] & L^2\,B \end{tikzcd}$$

which amounts to proving that for any $x : \Pi(q\,\alpha : p).\,A_q$ we have

$$\begin{aligned} (L^2\,f)_p\ ((\vartheta_{L\,A})_p\ x) &\overset{?}{=} (\vartheta_{L\,B})_p\ ((L\,f)_p\ x) \\ \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,f_r\ (\theta_{L\,A}\ \alpha\ x\ r\ \beta) &\overset{?}{=} \lambda(q\,\alpha : p).\,\theta_{L\,B}\ \alpha\ (\lambda(r\,\beta : p).\,f_r\ (x\ r\ \beta)) \\ \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,f_r\ (x\ r\ (\beta \circ \alpha)) &\overset{?}{=} \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,f_r\ (x\ r\ (\beta \circ \alpha)) \end{aligned}$$

First diagram We now turn to the diagram below.

$$\begin{tikzcd} 1 \arrow[r, "\vartheta"] \arrow[d, "\vartheta"'] & L \arrow[d, "\vartheta_L"] \\ L \arrow[r, "L\,\vartheta"] & L^2 \end{tikzcd}$$

This is equivalent to showing that for any $x : A_p$, we have

$$\begin{aligned} (\vartheta_{L\,A})_p\ ((\vartheta_A)_p\ x) &\overset{?}{=} (L\,\vartheta_A)_p\ ((\vartheta_A)_p\ x) \\ \lambda(q\,\alpha : p).\,\theta_{L\,A}\ \alpha\ (\lambda(q\,\alpha : p).\,\theta_A\ \alpha\ x) &\overset{?}{=} \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,\theta_A\ \beta\ (\theta_A\ \alpha\ x) \\ \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,\theta_A\ (\beta \circ \alpha)\ x &\overset{?}{=} \lambda(q\,\alpha : p).\,\lambda(r\,\beta : q).\,\theta_A\ \beta\ (\theta_A\ \alpha\ x) \end{aligned}$$

which is a direct consequence of the functoriality of θ_A.

Second diagram We focus on the following diagram.

$$\begin{tikzcd} 1 \arrow[r, "\vartheta"] \arrow[dr, "="'] & L \arrow[d, "\varepsilon"] \\ & 1 \end{tikzcd}$$

This is equivalent to showing that for any $x : A_p$, we have

$$\begin{aligned} (\varepsilon_A)_p\ ((\vartheta_A)_p\ x) &\stackrel{?}{=} x \\ \theta_A\ \mathtt{id}_p\ x &\stackrel{?}{=} x \end{aligned}$$

which is the other functoriality equation of θ_A.

Third diagram

$$\begin{tikzcd} L \arrow[r, "\vartheta_L"] \arrow[dr, "="'] & L^2 \arrow[d, "L\,\varepsilon"] \\ & L \end{tikzcd}$$

This is equivalent to showing that for any $x : \Pi(q\,\alpha : p).\,A_q$, we have

$$\begin{aligned} (L\,\varepsilon_A)_p\ ((\vartheta_{L\,A})_p\ x) &\stackrel{?}{=} x \\ \lambda(q\,\alpha : p).\,x\ q\ (\mathtt{id}_q \circ \alpha) &\stackrel{?}{=} x \end{aligned}$$

which is a consequence of $\mathbf{C}$ being a category.

3 E que s'apelerio Thunkable

The main result of this note lies here. Now that we proved that $\mathbf{C}^{\recycle}$ was a model of effectful call-by-value, let us have a look at the semantic notion of purity in this model. Without further ado:

Theorem 3. *Thunkable morphisms in $\mathbf{C}^{\recycle}$ are exactly natural transformations.*

Proof. Let $f : \mathbf{C}^{\recycle}(A, B)$. By definition, it is thunkable when the following diagram commutes.

$$\begin{tikzcd} A \arrow[r, "\vartheta_A"] \arrow[d, "f"'] & L\,A \arrow[d, "L\,f"] \\ B \arrow[r, "\vartheta_B"] & L\,B \end{tikzcd}$$

This is in turn equivalent to the following equation holding for any $x : A_p$.

$$\begin{aligned} (L\,f)_p\ ((\vartheta_A)_p\ x) &= (\vartheta_B)_p\ (f_p\ x) \\ \lambda(q\,\alpha : p).\,f_q\ (\theta_A\ \alpha\ x) &= \lambda(q\,\alpha : p).\,\theta_B\ \alpha\ (f_p\ x) \end{aligned}$$

Up to function extensionality, which holds in **Set**, this is literally the naturality equation for f. $\square$

Let us insist that while thunkability is a naturality equation in $\mathbf{C}^{✿}$, the naturality discussed above does not live at the same level. Said otherwise, a $\mathbf{C}^{✿}$-morphism is natural w.r.t. ϑ iff it is natural *as a presheaf morphism*, which are *a priori* two distinct notions.

A direct consequence of this theorem is the following observation.

Theorem 4. *The presheaf category $\hat{\mathbf{C}}$ is exactly the subcategory of $\mathbf{C}^{✿}$ restricted to thunkable morphisms.*

Insight 6. This justifies the claims from the abstract. Artificial presheaves form an abstract Kleisli category and thus a model of the call-by-value λ-calculus. By semantically restricting it to terms that behaves as if they were values, we get a model of the full λ-calculus, including arbitrary β-rule. This can be presented in the same syntactic way as the realizability trick described before.

The inclusion is not trivial, i.e. in general not all $\mathbf{C}^{✿}$-morphisms are thunkable. For instance, pick $\mathbf{C} := \mathbf{2}$, the two-point discrete category. Let consider the terminal object 1 and the coproduct $+$ in the presheaf category. Then, up to extensionality $|\hat{\mathbf{C}}(1, 1+1)| \cong 1+1$ but $|\mathbf{C}^{✿}(1, 1+1)| \cong 1+1 \to 1+1$.

Insight 7. This is saying that there are exactly two values of boolean type[2], but that there are four effectful booleans in this model. In addition to `true` and `false`, which correspond to the two constant functions of type $1+1 \to 1+1$, there are two terms that use the argument, namely the identity and the flip functions.

4 Curry-Howard interpretation

So now we have a direct style presentation of an effectful programming language. Can we understand the dynamic content of this effect? In this section we show that the description given in $\mathbf{C}^{✿}$ can be translated to one given in terms of **Set**.

Let us start by remarking that $\hat{\mathbf{C}}$ contains **Set** in the following sense. If $A : \mathbf{Set}$, we write $A^{\mathbf{C}} : \hat{\mathbf{C}}$ for the constant presheaf returning A. Then $\hat{\mathbf{C}}(A^{\mathbf{C}}, B^{\mathbf{C}}) \cong \mathbf{Set}(A, B)$. Due to their universal properties, the type formers in **Set** are isomorphic to their presheaf equivalent through this translation.

Insight 8. This means that $\hat{\mathbf{C}}$ is just a bigger version of **Set**. It has *a priori* more types, but on their common types they have the same terms.

We make formal the propery of being an object from **Set**.

Definition 9. We say that a presheaf is *pure* if it is isomorphic (in $\hat{\mathbf{C}}$) to a constant presheaf.

Considering the above isomorphism, it is easy to see that the subcategory of pure presheaves is equivalent to **Set**. If you remove the new types that have been added by

[2] No shit Sherlock!

the presheaf construction, you gained nothing, as long as you only consider effect-free morphisms. But what happens for effectful morphisms?

We go back to effects for a second, to get rid of the direct style presentation. Instead, we will canonically define a monad and work in its Kleisli category.

Proposition 5. *Given any abstract Kleisli category* $(\mathbf{K}, L, \vartheta, \varepsilon)$*, let us write* $\Theta(\mathbf{K})$ *for the subcategory of thunkable morphisms.* L *trivially induces a functor* $\Theta(L) : \Theta(\mathbf{K}) \to \Theta(\mathbf{K})$*. Furthermore,* $\Theta(L)$ *is a monad and its Kleisli category is equivalent to* $\mathbf{K}$*.*

Nothing involved there, it is almost by definition. But this means that we can also see the $L : \mathbf{C}^{\text{♻}} \to \mathbf{C}^{\text{♻}}$ from the previous section as a monad $\hat{L} : \hat{\mathbf{C}} \to \hat{\mathbf{C}}$. For pure presheaves, it turns out that the Kleisli morphisms in $\hat{\mathbf{C}}$ are equivalent to Kleisli morphisms for a specific reader monad in the **Set** category.

Definition 10. We define $\mathcal{S}$ the set of sieves of $\mathbf{C}$ as the quotient

$$\{(p : \mathbf{C}, q : \mathbf{C}, \alpha : \mathbf{C}(q, p))\} / \sim_{\mathcal{S}}$$

where

$$(p, q, \alpha) \sim_{\mathcal{S}} (r, s, \beta) := q = r.$$

That is, sieves are cones of morphisms that have the same target object.

Proposition 6. *Let* R *be the reader monad in* **Set** *over sieves, i.e.* $R\,A := \mathcal{S} \to A$*. We have the following isomorphism:*

$$\hat{C}(A^{\mathbf{C}}, \hat{L}\,B^{\mathbf{C}}) \cong \mathbf{Set}(A, R\,B).$$

Proof. Let $f : \hat{C}(A^{\mathbf{C}}, \hat{L}\,B^{\mathbf{C}})$. That is, we have a family of functions

$$f_p : A \to \Pi(q\,\alpha : p).\,B$$

that satisfy the following naturality equation for any $\alpha : \mathbf{C}(q, p)$.

$$\begin{tikzcd}
A \arrow[r, "="] \arrow[d, "f_p"'] & A \arrow[d, "f_q"] \\
\Pi(q\,\alpha : p).\,B \arrow[r, "\theta_{L\,B}\ \alpha"] & \Pi(q\,\alpha : p).\,B
\end{tikzcd}$$

Spelled out, this means that for any $x : A$, $\alpha : \mathbf{C}(q, p)$ and $\beta : \mathbf{C}(r, q)$,

$$f_q\ x\ r\ \beta = f_p\ x\ r\ (\beta \circ \alpha).$$

Now, simply define $\varphi : A \to \mathcal{S} \to B$ as

$$\varphi\ x\ (p, q, \alpha) := f_p\ x\ q\ \alpha$$

which by virtue of the above naturality requirement preserves the sieve quotient.

Dually, if $\varphi : A \to \mathcal{S} \to B$, define $f : \hat{C}(A^{\mathbf{C}}, \hat{L}\, B^{\mathbf{C}})$ as

$$f_p\ x\ q\ \alpha := \varphi\ x\ (p, q, \alpha).$$

It is trivial to check that the naturality equation holds. Furthermore, these two translations define indeed an isomorphism. $\square$

Unfortunately, if $A : \mathbf{Set}$, then in general $\hat{L}\, A^{\mathbf{C}}$ is not pure, which prevents seeing $\hat{L}$ as a monad over the subcategory of pure presheaves.

5 Other

For completeness, Führmann also defines weaker notions of purity, but it turns out that they are degenerate in $\mathbf{C}^{\text{♻}}$.

Proposition 7. *All morphisms in $\mathbf{C}^{\text{♻}}$ are copyable, discardable and central.*

This is similar to the case of the monad $T\ X := X \times X$, which is not surprising as the resulting Kleisli category is a full subcategory of the artificial presheaf model over the discrete two-point category $\mathbf{2}$.

References

[Füh99] Carsten Führmann. "Direct Models for the Computational Lambda Calculus". In: *Electr. Notes Theor. Comput. Sci.* 20 (1999), pp. 245–292. DOI: 10.1016/S1571-0661(04)80078-1. URL: https://doi.org/10.1016/S1571-0661(04)80078-1.

[Kle45] Stephen Cole Kleene. "On the Interpretation of Intuitionistic Number Theory". In: *J. Symb. Log.* 10.4 (1945), pp. 109–124. DOI: 10.2307/2269016. URL: https://doi.org/10.2307/2269016.

[Lev04] Paul Blain Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*. Vol. 2. Semantics Structures in Computation. Springer, 2004. ISBN: 1-4020-1730-8.

[Lev17] Paul Blain Levy. "Contextual isomorphisms". In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*. 2017, pp. 400–414. URL: http://dl.acm.org/citation.cfm?id=3009898.