Yoneda and CPS

October 22, 2024

Contents

This is a note based on a short conversation with Bartosz Milewski at ICFP 2024. Bartosz Milewski is the author of Category Theory for Programmers. I am grateful for his kindness and the short conversation.

1 Why I am Writing This

While at ICFP 2024, I asked Bartosz: "How is Yoneda Lemma used in the field of programming language". I got an answer along the line of:

Apply Yoneda Lemma to the identity functor gives you CPS

I shall attempt to illustrate my understanding of this concept

2 Background and Setup

2.1 Yoneda Lemma

We shall take Yoneda lemma for granted, which is stated as follow:

$$\forall F : \mathsf{C} \to \mathsf{Set}\,.\ \mathrm{Hom}(\mathsf{C}(c,-), F) \cong Fc$$

2.1.1 Meaning of the Symbols

C A locally small category

c An object in C

F A normal (a.k.a covariant) functor of the type C → Set

$\mathrm{Hom}(F, G)$ The collection of natural transformations from functor F to functor G

Fc The image of object $c \in \mathsf{C}$ mapped by F

$\cong$ Set isomorphism

2.2 Category of Types

We make the assumption the reader is familiar with the concept of "category of types" (Those without can refer to Steve Awodey's Textbook (Awodey, Steve, 2010), Chapter 6, Section 6)

We shall denote a category of types as $\mathsf{C}(\lambda)$

I also made the assumption that each type is concretely represented by a set of expressions with that type.

I think this is needed to make the "identity functor" part make sense

2.3 Natural Transformation in the Context of Category of Types

WARNING: For this section, I only have intuition, I very much may explain things wrong. I recommend reading more from more experienced people

- Bartosz's own blog
- *Theorems for Free* by Philip Wadler (Wadler, Philip, 1989)
- *Types, Abstraction and Parametric Polymorphism* by John C. Reynolds

Intuitively, a parametric function of the type `forall a .  F a` is:

A collection of functions, one for each type

This definition is similar to the definition of natural transformation:

A collection of morphism, one for each object

I believe, a parametric polymorphic function of the type `forall a .  F a \to G a` is in fact, a natural transformation between functor `F, G`

3 Main Theorem

Let us instantiate Yoneda Lemma with:

- $\mathsf{C} = \mathsf{C}(\lambda)$
- $F = I$
- $c = A$ for some type A

Our isomorphism now becomes

$$\mathrm{Hom}(\mathsf{C}(\lambda)(A, -), I) \cong IA = A$$

3.1 RHS

On the RHS, we have a type A, which is the set of all the expressions of type A

3.2 LHS

Consider an arbitrary natural transformation α in $\mathrm{Hom}(\mathsf{C}(\lambda)(A, -), I)$ We have the following commutative diagram w.r.t α

$$\begin{tikzcd}
\mathsf{C}(\lambda)(A, B) \arrow[r, "\alpha_B"] \arrow[d, "f_*"] & B \arrow[d, "If"] \\
\mathsf{C}(\lambda)(A, C) \arrow[r, "\alpha_C"] & C
\end{tikzcd}$$

We now notice:

- $\mathsf{C}(\lambda)(A, B)$ in the category of types is functions of type $A \to B$. (This is also the exponential $B^A \in \mathsf{C}(\lambda)$)

Thus we get:

$$\begin{tikzcd}
(A \to B) \arrow[r, "\alpha_B"] \arrow[d, "f_*"] & B \arrow[d, "If"] \\
(A \to C) \arrow[r, "\alpha_C"] & C
\end{tikzcd}$$

Thus each component of α has type:

$$\alpha_b : (A \to b) \to b$$

Consider α as a collection of functions, we get

$$\alpha : \forall b\,.\,(A \to b) \to b$$

Which is the type of CPS.

3.3 Isomorphism

Bringing back isomorphism between LHS and RHS, we get:

The collection of CPS functions of type

$$\forall b\,.\,(A \to b) \to b$$

is isomorphic to the collection of expressions of type

$$A$$