Recap:

The Yoneda lemma tells us that $\forall$ functor $F: C \to Set$ and $\forall a \in C$, it holds that

$$Nat(C(a,-), F) \simeq Fa$$

One possible bijection is

$$\phi : Nat(C(a,-), F) \to Fa$$
$$\phi \{\alpha_x\}_x = \alpha_a \, id_a$$

Moreover, it gives us a recipe for enumerating all natural transformations

$$\{\alpha_x\}_x : C(a,-) \Rightarrow F$$

by setting

$$\alpha_x = \lambda f. (F f)\, p$$

where $p = \alpha_a \, id_a$ can be whatever.

In Haskell, this means that

$$\forall x. (a \to x) \to F x \simeq F a$$

① Show that

$$\phi :: (\forall x.\ (a \to x) \to F\,x) \to F\,a$$

$$\phi\ \alpha_x = \alpha_a\ id_a$$

and

$$\psi :: F\,a \to \forall x.\ (a \to x) \to F\,x$$

$$\psi\ u\ h_x = (F\ h_x)\ u$$

are inverses of one another.

Solution:

$$\begin{aligned}
(\phi \circ \psi)\,u &= \phi\ \lambda h_x.(F\ h_x)\ u \\
&= (\lambda\ h_x.\ F\,h_x\ u)\ id_a \\
&= F\ id_a\ u \\
&= id_{Fa}\ u \\
&= u \quad \checkmark
\end{aligned}$$

$$\begin{aligned}
(\psi \circ \phi)\ \alpha_x &= \psi\ (\alpha_a\ id_a) \\
&= \lambda\ h_x.\ (F\ h_x)\ (\alpha_a\ id_a) \\
&\overset{\text{Yoneda}}{=} \alpha_x \quad \checkmark
\end{aligned}$$

$\square$

(2)

What does the Yoneda lemma tell us when applied to a discrete category?

Solution:

It tells us that there are $\# Fa$ natural transformations

$$C(a,-) \Rightarrow F$$

Indeed, take an $x \neq a$ in C. Then $C(a,x) = \emptyset$. Thus, there is exactly 1 function $\emptyset \to Fu$ $\forall u$, namely the empty function $\emptyset$. Consider now a map $\tau_a : C(a,a) \to Fa$. By the naturality condition, we only need to check what happens to $\tau_a\, id_a$. There are $\# Fa$ choices for it, corroborating the claim from the lemma.

(3)

Construct another representation of [()] using the Yoneda lemma.

Solution:

By the Yoneda lemma we know that

$$\forall x.\, (() \to x) \to [x] \simeq [()]$$

So any f with that signature will work.

Intuitively, a map $() \to X$ is just a container for an X, which is then repeated in a list.