Yoneda (Proof)

To prove:

$$F(C) \simeq_{Set} [\mathcal{C}, Set](\mathcal{C}(C, -), F)$$

nat. in F, C

we omit to prove this (see e.g. Awodey)

Let $\eta : \mathcal{C}(C, -) \longrightarrow F$ nat. tr.

by naturality

$$\forall X, Y \in |\mathcal{C}|.\ \forall h : X \to_{\mathcal{C}} Y.\ \mathcal{C}(C, h); \eta_Y = \eta_X ; Fh \quad (\text{in } Set)$$

i.e.

$$\forall X, Y \in |\mathcal{C}|.\ \forall h : X \to_{\mathcal{C}} Y.\ \forall g : C \to_{\mathcal{C}} X.\ \eta_Y(\mathcal{C}(C, h)(g)) = Fh(\eta_X(g))$$

i.e. $\forall \ldots$

$$\eta_Y(g ; h) = Fh(\eta_X(g))$$

In particular, choosing $X = C$, $g = id_C : C \to X = C$

$$\forall Y \in |\mathcal{C}|.\ \forall h : C \to Y.\ \eta_Y(h) = Fh(\eta_C(id_C)) \quad (*)$$

Now, let's define the iso:

$$f : F(C) \longrightarrow_{Set} [\mathcal{C}, Set](\mathcal{C}(C, -), F)$$

$$x \longmapsto \eta^x \quad \text{where}$$

$$\eta^x_A : \mathcal{C}(C, A) \longrightarrow_{Set} F(A)$$

$$h \longmapsto Fh(x)$$

Fh: $FC \to FA$ in Set; x: FC

(& : verify η natural)

$$f^{-1} : [\mathcal{C}, Set](\mathcal{C}(C, -), F) \longrightarrow_{Set} F(C)$$

$$\eta \longmapsto \eta_C(id_C)$$

η_C: $\mathcal{C}(C, C) \to FC$ in Set; id_C: $\mathcal{C}(C, C)$

Let's verify that f, f^{-1} are inverses.

$f; f^{-1} \overset{?}{=} id_{FC}$ (in Set)

Let $x \in FC$

$f^{-1}(f(x))$ [def f]

$= f^{-1}(\eta^x)$ [def f^{-1}]

$= \eta^x_C(id_C)$ [def η^x]

$= F id_C(x)$ [F functor]

$= id_{FC}(x) = x$

Now: $f^{-1}; f = id_{[\mathcal{C}, Set]}(\text{----})$ in Set

Let $\eta : \mathcal{C}(C, -) \to F$ nat tr.

$f(f^{-1}(\eta))$ [def f^{-1}]

$= f(\eta_C(id_C))$ [def f]

$= \eta^{(\eta_C(id_C))} =: \psi$ where

$\psi_A(h) = Fh(\eta_C(id_C))$

$\quad \| \text{ by } (*)$

$\quad \eta_A(h)$

$(\Rightarrow \psi = \eta)$

$= \eta$

Ex: a functor $F : \mathcal{C} \to \mathcal{D}$ is <u>full</u> iff $\forall A, B \in |\mathcal{C}|$

$\mathcal{C}(A, B) \xrightarrow[Set]{} \mathcal{D}(FA, FB)$

$h \longmapsto Fh$

is surjective

a functor F is <u>faithful</u> iff the above function is injective

Prove that $F : \mathcal{C} \to [\mathcal{C}, Set]$

$A \longmapsto \mathcal{C}(A, -)$

is fully faithful

(use Yoneda)

Ex: prove $(\forall \alpha.\ \alpha \to \tau) \simeq \tau$

$$
\begin{aligned}
&(\forall \alpha.\ \alpha \to \tau) \\
\simeq\ &(\forall \alpha.\ (1 \to \alpha) \to \tau) \\
\simeq\ &(\forall \alpha.\ (1 \to \alpha) \to \sigma(\alpha)) \\
&\qquad \text{with } \sigma(\alpha) \equiv \tau \\
\simeq\ &\sigma(1) = \tau
\end{aligned}
$$

$$
\begin{aligned}
&(\forall \alpha.\ \alpha \to \alpha) \stackrel{?}{\simeq} 1 \\
&\quad \simeq \forall \alpha.\ (1 \to \alpha) \to \sigma(\alpha) \\
&\qquad \text{with } \sigma(\alpha) \equiv \alpha \\
&\qquad \simeq \sigma(1) = 1
\end{aligned}
$$

$$
\begin{aligned}
&(\forall \alpha.\ \alpha \to \alpha \to \alpha) \stackrel{?}{\simeq} 2 = 1 + 1 \\
&\simeq \forall \alpha.\ (\alpha \times \alpha) \to \alpha \\
&\simeq \forall \alpha.\ (2 \to \alpha) \to \alpha \\
&\simeq 2
\end{aligned}
$$

Recursion in types

Motivation:

$$\mathbb{N} \simeq 1 + \mathbb{N}$$

$$\text{List}\,\mathbb{N} \simeq 1 + (\mathbb{N} \times \text{List}\,\mathbb{N})$$

$$\text{Tree}\,\mathbb{N} \simeq 1 + (\mathbb{N} \times \text{Tree}\,\mathbb{N} \times \text{Tree}\,\mathbb{N})$$

o o o

Two approaches:

- equi-recursive types
  e.g. $\mathbb{N}$ and $1+\mathbb{N}$ are <u>equal</u>
- iso-recursive types
  e.g. $\mathbb{N}$ and $1+\mathbb{N}$ are <u>isomorphic</u>
  in the progr. lang. we can use
  $f : \mathbb{N} \to 1+\mathbb{N}$
  $f^{-1} : 1+\mathbb{N} \to \mathbb{N}$ iso$_f$

Rec. types can be defined through
- recursive equations
- (type-level) fixed points

Note: for Curry-Howard, we need to forbid certain recursive types
e.g. $\tau \simeq (\tau \to 0)$
$P \leftrightarrow \neg P$

for instance: we only allow <u>covariant</u> recursion

(See practical examples)