1 Y-combinator homework comment

The important thing to know when typing the Y-combinator example is to carefully follow the rules or steps for doing all the typing.

2 Interpretation of equality

2.1 First Interpretation

\[
\begin{align*}
\text{Base Case} \\
\text{int} = \text{int} \\
\text{t}_1 = \text{t}_3 \quad \text{t}_2 = \text{t}_4 \\
\text{t}_1 \rightarrow \text{t}_2 = \text{t}_3 \rightarrow \text{t}_4 \\
\text{t}_1 [\alpha := \mu \cdot \text{t}_1] = \text{t}_2 [\alpha := \mu \cdot \text{t}_2] \\
\mu \alpha . \text{t}_1 = \mu \alpha . \text{t}_2 \\
\mu \alpha . \mu \beta . \alpha \rightarrow \beta \Rightarrow \mu \alpha . \mu \beta . \beta \rightarrow \alpha
\end{align*}
\]

Two terms are equal if you can decide the equality of two potentially bigger terms. Why is this a problem? Because of infinite unfolding.

2.2 Second Interpretation

\[
\begin{align*}
\alpha \in \Gamma \\
\Rightarrow \alpha = \alpha & \quad \text{Base Case} \\
\Gamma \vdash \text{int} = \text{int} & \quad \text{Base Case} \\
\Gamma \vdash \text{t}_1 = \text{t}_3 \quad \Gamma \vdash \text{t}_2 = \text{t}_4 \\
\Gamma \vdash \text{t}_1 \rightarrow \text{t}_2 = \text{t}_3 \rightarrow \text{t}_4 \\
\Gamma, \alpha \vdash \text{t}_1 = \text{t}_2 \\
\mu \alpha . \text{t}_1 = \mu \alpha . \text{t}_2
\end{align*}
\]
Here the equality decision will stop. Usually we see $\mu \alpha.\mu \beta.(\alpha \to \beta) + \text{int}$ written as $t ::= \text{int} \mid t \to t$ (not reflected in term language). What if we don’t want implicit equivalence of recursive types?

### 2.3 Third Interpretation

Put annotations in term types:

\[
\begin{align*}
\Gamma \vdash e : t[\alpha ::= \mu \alpha.t] & \quad \text{Introduction} \\
\Gamma \vdash e : \mu \alpha.t & \quad \text{Elimination}
\end{align*}
\]

The Elimination rule essentially says, “you see a $\mu$ type. From now on deal with this type as an unfolded version”. Now we do equality interpretation only when the programmer says “do it here and only once”. $(\mu \alpha.\alpha \to \alpha) \to (\mu \alpha.\alpha \to \alpha)$ is the result of substituting $(\mu \alpha.\alpha \to \alpha)$ into itself.