Comp 411 Notes

Recursive types
Scribe: Roumen Kaiabachev

March 31, 2005

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

$$\frac{}{int = int}\texttt{Base Case}$$

$$\frac{t_1 = t_3 \quad t_2 = t_4}{t_1 \rightarrow t_2 = t_3 \rightarrow t_4}$$

$$\frac{t_1\ [\alpha := \mu.t_1] = t_2\ [\alpha := \mu.t_2]}{\mu\alpha.t_1 = \mu\alpha.t_2}$$

$$\mu\alpha.\mu\beta.\alpha \rightarrow \beta\ ?=?\ \mu\alpha.\mu\beta.\beta \rightarrow \alpha$$

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

$$\frac{\alpha \in \Gamma}{\Gamma \vdash \alpha = \alpha}\texttt{Base Case}$$

$$\frac{}{\Gamma \vdash int = int}\texttt{Base Case}$$

$$\frac{\Gamma \vdash t_1 = t_3 \quad \Gamma \vdash t_2 = t_4}{\Gamma \vdash t_1 \rightarrow t_2 = t_3 \rightarrow t_4}$$

$$\frac{\Gamma, \alpha \vdash t_1 = t_2}{\mu\alpha.t_1 = \mu\alpha.t_2}$$

$$\mu\alpha.\mu\beta.(\alpha \to \beta) + int \;?=?\; \mu\alpha.\mu\beta.(\beta \to \alpha) + int$$

Here the equality decision will stop. Usually we see $\mu\alpha.\mu\beta.(\alpha \to \beta) + int$ written as $t ::= int \mid t \to t$ (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:

$$\frac{\Gamma \vdash e : t[\alpha := \mu\alpha.t]}{fold\, e : \mu\alpha.t}\texttt{Introduction}$$

$$\frac{\Gamma \vdash e : \mu\alpha.t}{unfold\, e : t[\alpha : \mu\alpha.t]}\texttt{Elimination}$$

The Elimination rule essentially says, "you see a μ 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.