# 2-15-06: One more case for SN; Boolean encodings; Girard's J operator

Quick case for FTLR for System F is SN:

*Case:*

$$\frac{\Delta;\Gamma \vdash e : \forall\alpha.\tau \qquad \Delta \vdash \sigma \text{ type}}{\Delta;\Gamma \vdash e[\sigma] : \tau[\sigma/\alpha]}$$

By induction, $\delta_T(\gamma(e)) \in \mathcal{C}[\![\forall\alpha.\tau]\!]\delta$. By the logical relation, $\delta_T(\gamma(e))[\delta_T(\sigma)] = \delta_T(\gamma(e[\sigma])) \in \mathcal{C}[\![\tau]\!]\delta, \alpha \mapsto (\delta_T(\sigma), R)$ with $R = \mathcal{C}[\![\sigma]\!]\delta$. We need:

**Lemma** If $\Delta \vdash \sigma$ type and $\delta \in \mathcal{D}[\![\Delta]\!]$ then $\mathcal{C}[\![\sigma]\!]\delta \in$ Cand.

*Proof:* by the three lemmas.

## More examples

**true** $= \Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.x$
**false** $= \Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.y$

**Theorem** $\forall\tau.\forall v_1, v_2 : \tau,\ f[\tau](v_1)(v_2) \in \{v_1, v_2\}$

$f \in \mathcal{V}[\![\forall\alpha.\alpha \to (\alpha \to \alpha)]\!]$. Pick $\sigma = \tau$, $R = \{v_1, v_2\}$. Then $f[\tau] \in \mathcal{C}[\![\alpha \to (\alpha \to \alpha)]\!]\alpha \mapsto (\tau, R)$. Hence $f[\tau] \downarrow v \in \mathcal{V}[\![\alpha \to \alpha \to \alpha]\!]\alpha \mapsto (\tau, R)$. Since $v_1 \in R$, $v(v_1) \in \mathcal{C}[\![\alpha \to \alpha]\!]\alpha \mapsto (\tau, R)$. Hence $f[\tau](v_1) \downarrow w_1 \in \mathcal{V}[\![\alpha \to \alpha]\!]\alpha \mapsto (\tau, R)$. Since $v_2 \in R$, $w_1(v_2) \in \mathcal{C}[\![\alpha]\!] \ldots$. Hence $w_1(v_2) \downarrow w_2 \in R$. So $f[\tau](v_1)(v_2) \downarrow w_2 \in R = \{v_1, v_2\}$.

## Girard's $J$ operator

Girard's $J$ is a nonparametric operator (it allows you to "inspect" a type variable) $\approx$ intensional type analysis.

(This discussion will be in the full-reduction language with no base types)

$J : \forall\alpha.\forall\beta.\alpha \to \beta$

$$\begin{aligned} J[\sigma][\tau]e &\mapsto e && \text{if } \sigma = \tau \\ J[\sigma][\tau]e &\mapsto 0[\tau] && \text{if } \sigma \neq \tau, \sigma, \tau \text{ closed} \end{aligned}$$

$0 : \forall\alpha.\alpha$

$$\begin{aligned} 0[\sigma \to \tau]e &\mapsto 0[\tau] \\ 0[\forall\alpha.\tau]e &\mapsto 0[\tau[\sigma/\alpha]] \end{aligned}$$

In untyped lambda caluclus, you have $\omega = (\lambda x.xx)(\lambda x.xx)$. Then $\omega \mapsto \omega$.

Let $\rho = \forall\alpha.\alpha \to \alpha$.
$F : \rho = \Lambda\alpha.J[\rho \to \rho][\alpha \to \alpha](\lambda x : \rho.x[\rho]x)$
$\omega = F[\rho]F \mapsto^* F[\rho]F$

## Harper-Mitchell $J'$ operator

$J' : \forall\alpha.\forall\beta.(\alpha \to \alpha) \to (\beta \to \beta)$

$$\begin{array}{lcll} J'[\sigma][\tau]e & \mapsto & e & \text{if } \sigma = \tau \\ J'[\sigma][\tau]e & \mapsto & \lambda x : \tau.x & \text{if } \sigma \neq \tau, \sigma, \tau \text{ closed} \end{array}$$

Let $\rho = \forall\alpha.\alpha \to \alpha$.
$F : \rho = \Lambda\alpha.J'[\rho][\alpha](\lambda x : \rho.x[\rho]x)$
$\omega = F[\rho]F \mapsto^* F[\rho]F$

$J'$ cannot give you a fixpoint combinator:

Suppose you could write (using $J'$) a fixpoint combinator $Y : \forall\alpha.(\alpha \to \alpha) \to \alpha$. (So that $Y(f) \equiv f(Yf))$. Pick an uninhabited $\tau$ in System F. Then $Y[\tau](\lambda x.x) : \tau$. Now $\tau$ is inhabited. If you can encode $Y$ using $J'$ then you can encode $Y' : (\alpha \to \alpha) \to \alpha$ where $Y' = Y[\Lambda\alpha.\Lambda\beta.\lambda x.\lambda y.y/J']$. Then $Y'$ is written entirely in pure System F! And yet, $Y'[\tau](\lambda x.x)$ has type $\tau$. Contradiction.