

1 Background

My proof uses a denotation function mapping kinds to the set of normalizing type expressions of that kind as follows:

$$\begin{aligned}\llbracket * \rrbracket_\Gamma &= \{ \langle H, \tau \rangle \} \\ \llbracket A..B \rrbracket_\Gamma &= \{ \langle H, \tau \rangle \mid \Gamma \vdash A \leq \tau : *, \Gamma \vdash \tau \leq B : *, \} \\ \llbracket \Pi(X : J).K \rrbracket_\Gamma &= \{ \langle H, \lambda(X : J).A \rangle \mid \forall \tau_x \in \llbracket J \rrbracket_\Gamma. \langle H(X \mapsto \tau_x), A \rangle \in \mathcal{E}\llbracket K[\tau_x/X] \rrbracket_\Gamma \} \\ \mathcal{E}\llbracket K \rrbracket_\Gamma &= \{ \langle H, A \rangle \mid \exists \tau. H \vdash A \Downarrow \tau \wedge \langle H, \tau \rangle \in \llbracket K \rrbracket_\Gamma \}\end{aligned}$$

so strong normalization can be stated as

$$\frac{\Gamma \vdash A : K \quad \Gamma \models H}{\langle H, A \rangle \in \mathcal{E}\llbracket K \rrbracket_\Gamma} \text{STRONG-NORMALIZE}$$

which is proven by induction on the judgment $\Gamma \vdash A : K$.

2 The Issue

Consider the variable case $A = X$, with the relevant rules being

$$\frac{\Gamma \text{ ctx} \quad \Gamma(X) = K}{\Gamma \vdash X : K} \text{K-VAR} \qquad \frac{H(X) = \tau}{H \vdash X \Downarrow^0 \tau} \text{EVAL-VAR}$$

To show that $\langle H, X \rangle \in \llbracket K \rrbracket_\Gamma$, we need to show $\tau \in \llbracket K \rrbracket_\Gamma$. Even if we assume a strict call-by-value semantics, we only know that $\Gamma \vdash \tau : K$ and $H \vdash \tau \text{ val}$.

One solution would be to bake in $\llbracket \cdot \rrbracket_\Gamma$ to the definition of the heap H , but that feels like a mistake.

So we need to prove

$$\frac{\Gamma \vdash \tau : K \quad H \vdash \tau \text{ val}}{\langle H, \tau \rangle \in \llbracket K \rrbracket_\Gamma} \text{DENOT-SPEC}$$

Ideally, this would be trivial, as the entire point of $\llbracket \cdot \rrbracket_\Gamma$ is for this to be true, and it is a mostly straightforward induction on the judgment $H \vdash \tau \text{ val}$. However, we run into an issue in the case that $\tau = \lambda(X : K).A$, with relevant rules/clauses:

$$\begin{aligned}&\overline{H \vdash \lambda(X : K).A \text{ val}} \\ \llbracket \Pi(X : J).K \rrbracket_\Gamma &= \{ \langle H, \lambda(X : J).A \rangle \mid \forall \tau_x \in \llbracket J \rrbracket_\Gamma. \langle H(X \mapsto \tau_x), A \rangle \in \\ &\quad \mathcal{E}\llbracket K[\tau_x/X] \rrbracket_\Gamma \}\end{aligned}$$

The issue arises when showing that $\langle H(X \mapsto \tau_x), A \rangle \in \mathcal{E}[[K]]_\Gamma$. Intuitively, this is simple; we just use the main strong normalization proof. But now we're doing some mutual induction that I'm not convinced is well-founded – we invoke STRONG-NORMALIZE with A (which is smaller than $\lambda(X : K).A$) and $H, X = \tau_x$ (which is larger than H). But then the proof of STRONG-NORMALIZE invokes DENOT-SPEC with an *arbitrary* τ in the var case.

My best guess is that I need to somehow use some measure of the heap size and term size combined, but I don't know if that actually works – τ can be arbitrarily large, and we don't shrink the heap when we pass it back to DENOT-SPEC.