

EVALUATION SEMANTICS OF COERCION

JONATHAN STERLING

An idea for a *new* evaluation dynamics of coercion inspired by the denotational semantics of cubical type theory. The main idea is that coercions should *not* be evaluated in an ordinary environment, but instead an environment where each cell is a line.

$$\begin{aligned}
& \llbracket \Gamma \vdash A \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \mathbf{Type} \\
& \llbracket \Gamma \vdash a : A \rrbracket : \prod_{\gamma : \llbracket \Gamma \rrbracket} \llbracket \Gamma \vdash A \rrbracket_{\gamma} \\
& \mathbf{coe} \llbracket \Gamma \vdash A \rrbracket : \prod_{\gamma : \llbracket \Gamma \rrbracket^i} \prod_{r, s : \mathbb{I}} \prod_{a : \llbracket \Gamma \vdash A \rrbracket_{\gamma(r)}} \llbracket \Gamma \vdash A \rrbracket_{\gamma(s)} \\
& \llbracket \Gamma \vdash \mathbf{coe}_F^{r \rightarrow s} M : F(r) \rrbracket_{\gamma} = \mathbf{coe} \llbracket \Gamma . \mathbb{I} \vdash F(i) \rrbracket_{\langle \mathbf{refl}(\gamma), \mathbf{id}_{\mathbb{I}} \rangle} \llbracket \Gamma \vdash r : \mathbb{I} \rrbracket_{\gamma} \rightarrow \llbracket \Gamma \vdash s : \mathbb{I} \rrbracket_{\gamma} \llbracket \Gamma \vdash M : F(r) \rrbracket_{\gamma} \\
& \mathbf{coe} \llbracket \Gamma \vdash \Pi(A, B) \rrbracket_{\gamma}^{r \rightarrow s} f = \lambda a. \mathbf{coe} \llbracket \Gamma . A \vdash B \rrbracket_{\langle \gamma, \lambda i. \mathbf{coe} \llbracket \Gamma \vdash A \rrbracket_{\gamma}^{s \rightarrow i} a \rangle} \mathbf{coe} \llbracket \Gamma \vdash A \rrbracket_{\gamma}^{s \rightarrow r} a \\
& \mathbf{coe} \llbracket \Gamma \vdash \Sigma(A, B) \rrbracket_{\gamma}^{r \rightarrow s} p = (\mathbf{coe} \llbracket \Gamma \vdash A \rrbracket_{\gamma}^{r \rightarrow s} \pi_1(p), \mathbf{coe} \llbracket \Gamma . A \vdash B \rrbracket_{\langle \gamma, \lambda i. \mathbf{coe} \llbracket \Gamma \vdash A \rrbracket_{\gamma}^{r \rightarrow i} \pi_1(p) \rangle} \pi_2(p)) \\
& \mathbf{coe} \llbracket \Gamma \vdash \mathbf{V}_i(A, B, E) \rrbracket_{\gamma}^{r \rightarrow s} (v) = \begin{cases} \mathbf{coe} \llbracket \Gamma, i = 0 \vdash A \rrbracket_{\langle \gamma, \star \rangle}^{r \rightarrow s} (v) & \text{if } \forall j, \llbracket i \rrbracket_{\gamma(j)} = 0 \\ \mathbf{coe} \llbracket \Gamma \vdash B \rrbracket_{\gamma}^{r \rightarrow s} (v) & \text{if } \forall j, \llbracket i \rrbracket_{\gamma(j)} = 1 \\ ? & \text{if } \forall j, \llbracket i \rrbracket_{\gamma(j)} = j \\ \mathbf{Vin}_{\llbracket i \rrbracket_{\gamma(s)}} (\mathbf{coe} \llbracket \Gamma . i = 0 \vdash A \rrbracket_{\langle \gamma, \star \rangle}^{r \rightarrow s} v, \mathbf{com} \llbracket \Gamma \vdash B \rrbracket_{\gamma}^{r \rightarrow s} [\partial(\llbracket i \rrbracket_{\gamma(s)})](\dots)) & \text{otherwise} \end{cases}
\end{aligned}$$

The part I don't know how to do is to evaluate anything that isn't a type constructor!! LOL