Note that as a matter of convention, I write $(s\ \Phi)$ to mean "delete this tube if the resulting instantiation mentions Φ" (it's how I implement $\forall$).

CoeFcom/o

$$\frac{(s\ r') = (s'\ r') \qquad (s\ x) \neq (s'\ x) \qquad (s_i\ x) \neq (s_i'\ x)}{(\mathsf{coe}^{r \rightsquigarrow r'}_{[x]\ (\mathsf{fcom}^{(s\ x) \rightsquigarrow (s'\ x)}\ (A\ x)\ \overrightarrow{[(s_i\ x) = (s_i'\ x) \hookrightarrow [z]\ (B_i\ x\ z)]})\ \mapsto}\ M)}{(\mathsf{gcom}^{r \rightsquigarrow r'}_{[x]\ (A\ x)}\ M\ \overrightarrow{[(s_i\ \Phi) = (s_i'\ \Phi) \hookrightarrow [x]\ (\mathsf{coe}^{(s'\ x) \rightsquigarrow (s\ x)}_{[z]\ (B_i\ x\ z)}\ (\mathsf{coe}^{r \rightsquigarrow x}_{[x]\ (B_i\ x\ (s\ x))}\ M))]}\ [(s\ \Phi) = (s'\ \Phi) \hookrightarrow [x]\ (\mathsf{coe}^{r \rightsquigarrow x}_{[x]\ (A\ x)}\ M)])}$$

It is not at all clear how to implement the evaluation semantics for such a rule, at least in a naïve way. That is, in the presentation above we maintain the fiction that we can "see" into the binder-closures well enough to take out the $(A\ x)$ (or even the $(s\ x)$!), etc. and use it elsewhere. We do not have this luxury, however, in the semantic domain; generally, the closures need to be thought of as black boxes.

One idea that doesn't work is to add a new kind of stack frame to the binder closure, that says *During instantiation, if this becomes an* fcom*, project the cap!* But this doesn't work, since depending on with what dimension the closure is instantiated, the fcom may disappear, and (for instance) be replaced by one of its tubes. Then, we have lost track of the cap forever.

However, I think there is a way out. During evaluation, when we encounter a non-rigid composition, we currently project out the appropriate part (either the cap or one of the tubes), and lose the rest of the information. We could instead *not* do this, but actually just annotate the composition with the value that would have been projected out.

Then, meta-operations like semantic function application would see and use this projection; likewise, quotation and definitional equivalence would do the same. However, by keeping this information, we would be able to extend the language of closures with a stack frame that can project the cap from an fcom, even if at instantiation-time it would have already evaluated to the tube. In particular, this allows us to implement a continuation which grabs the appropriate part of the composition for use in the resulting gcom.