

Computations

$$\begin{aligned}
\llbracket V \ W \rrbracket &= \bar{\lambda} \kappa s. \llbracket V \rrbracket \underline{@} \llbracket W \rrbracket \underline{@} \downarrow \kappa s \\
\llbracket V \ T \rrbracket &= ??? \\
\llbracket \mathbf{absurd} \ V \rrbracket &= \bar{\lambda} \kappa s. \mathbf{absurd} \llbracket V \rrbracket \\
\llbracket \mathbf{return} \ V \rrbracket &= \bar{\lambda} \kappa \ :: \ \kappa s. \kappa \underline{@} \llbracket V \rrbracket \underline{@} \downarrow \kappa s \\
\llbracket \mathbf{let} \ x \leftarrow M \ \mathbf{in} \ N \rrbracket &= \bar{\lambda} \kappa \ :: \ \kappa s. \llbracket M \rrbracket \overline{@} \left((\underline{\lambda} x \ \kappa s. \llbracket N \rrbracket \overline{@} (\kappa \ :: \ \uparrow \kappa s)) \ :: \ \kappa s \right) \\
\llbracket \mathbf{do} \ \ell \ V \rrbracket &= \bar{\lambda} \kappa \ :: \ \chi \ :: \ \eta \ :: \ \kappa s. \eta \underline{@} (\ell \langle \llbracket V \rrbracket, \eta \ :: \ \kappa \ :: \ [] \rangle) \underline{@} \downarrow \kappa s \\
\llbracket \mathbf{handle} \ M \ \mathbf{with} \ H \rrbracket &= \bar{\lambda} \kappa \ :: \ \chi \ :: \ \kappa s. \llbracket M \rrbracket \overline{@} (\llbracket H^{\text{ret}} \rrbracket \ :: \ \chi \ :: \ \llbracket H^{\text{ops}} \rrbracket \ :: \ \kappa \ :: \ \chi \ :: \ \kappa s), \text{ where}
\end{aligned}$$

$$\begin{aligned}
\llbracket \mathbf{return} \ x \mapsto N \rrbracket &= \underline{\lambda} x \ \kappa s. \underline{\mathbf{let}} \ (kx \ :: \ kf \ :: \ \kappa s') = \kappa s \ \underline{\mathbf{in}} \\
&\quad \llbracket N \rrbracket \overline{@} \uparrow \kappa s' \\
\llbracket \ell \ p \ r \mapsto N_\ell \rrbracket &= \underline{\lambda} (\ell \langle p, s \rangle) \ \kappa s. \underline{\mathbf{let}} \ r = \mathbf{fun} \ s \ \underline{\mathbf{in}} \\
&\quad \llbracket N_\ell \rrbracket \overline{@} \uparrow \kappa s
\end{aligned}$$

$$\begin{aligned}
\llbracket \mathbf{raise} \ V \rrbracket &= \bar{\lambda} \kappa \ :: \ \chi \ :: \ \eta \ :: \ \kappa s. \chi \underline{@} \llbracket V \rrbracket \underline{@} \downarrow \kappa s \\
\llbracket \mathbf{try} \ M \ \mathbf{with} \ H \rrbracket &= \bar{\lambda} (\kappa \ :: \ \chi \ :: \ \eta \ :: \ \kappa s \ \mathbf{as} \ \kappa s'). \llbracket M \rrbracket \overline{@} (K_{\text{ret}} \ :: \ \llbracket H \rrbracket \ :: \ \eta \ :: \ \kappa s'), \text{ where}
\end{aligned}$$

$$\begin{aligned}
K_{\text{ret}} &= \underline{\lambda} x \ \kappa s. \underline{\mathbf{let}} \ (kx \ :: \ kf \ :: \ k \ :: \ \kappa s') = \kappa s \ \underline{\mathbf{in}} \\
&\quad k \underline{@} x \underline{@} \kappa s' \\
\llbracket e \mapsto N \rrbracket &= \underline{\lambda} e \ \kappa s. \llbracket N \rrbracket \overline{@} \uparrow \kappa s
\end{aligned}$$

Top level program

$$\top \llbracket M \rrbracket = \llbracket M \rrbracket \overline{@} ((\underline{\lambda} x \ \kappa s. x) \ :: \ (\underline{\lambda} z \ \kappa s. \mathbf{absurd} \ z) \ :: \ \uparrow [])$$