

Regular machine: global environments

%datatype *clos*

%datatype *c-env*

%datatype *k-env*

%datatype *stack*

%name *clos* *c*

%name *c-env* \mathcal{E}

%name *k-env* \mathcal{E}_μ

%name *stack* \mathcal{S}

$c \quad ::= \quad (t, \mathcal{E}, \mathcal{E}_\mu)$

$\mathcal{E} \quad ::= \quad ()$
 $\quad \quad \quad | \quad (c; \mathcal{E})$

$\mathcal{E}_\mu \quad ::= \quad ()$
 $\quad \quad \quad | \quad (\mathcal{S}; \mathcal{E}_\mu)$

$\mathcal{S} \quad ::= \quad []$
 $\quad \quad \quad | \quad c :: \mathcal{S}$

%datatype *state*

%name *state* σ

$\sigma \quad ::= \quad \langle t, \mathcal{E}, \mathcal{E}_\mu, \mathcal{S} \rangle$

Regular machine: evaluation rules

%judgment $\sigma_1 \rightsquigarrow \sigma_2$

$$\langle n, \mathcal{E}, \mathcal{E}_\mu, \mathcal{S} \rangle \rightsquigarrow \langle t, \mathcal{E}', \mathcal{E}'_\mu, \mathcal{S} \rangle$$

[k·var] *when* $\mathcal{E}(n) = (t, \mathcal{E}', \mathcal{E}'_\mu)$

$$\langle (t \ u), \mathcal{E}, \mathcal{E}_\mu, \mathcal{S} \rangle \rightsquigarrow \langle t, \mathcal{E}, \mathcal{E}_\mu, (u, \mathcal{E}, \mathcal{E}_\mu) :: \mathcal{S} \rangle$$

[k·app]

$$\langle \lambda t, \mathcal{E}, \mathcal{E}_\mu, c :: \mathcal{S} \rangle \rightsquigarrow \langle t, (c; \mathcal{E}), \mathcal{E}_\mu, \mathcal{S} \rangle$$

[k·abs]

$$\langle \mathbf{catch} \ t, \mathcal{E}, \mathcal{E}_\mu, \mathcal{S} \rangle \rightsquigarrow \langle t, \mathcal{E}, (\mathcal{S}; \mathcal{E}_\mu), \mathcal{S} \rangle$$

[k·catch]

$$\langle \mathbf{throw} \ \alpha \ t, \mathcal{E}, \mathcal{E}_\mu, \mathcal{S} \rangle \rightsquigarrow \langle t, \mathcal{E}, \mathcal{E}_\mu, \mathcal{S}' \rangle$$

[k·throw] *when* $\mathcal{E}_\mu(\alpha) = \mathcal{S}'$

%unique $+\sigma_1 \rightsquigarrow -1\sigma_2$