1 Krivine abstract machine

1.1 Syntax

$unary$: **type**.
$term$: **type**.
$clos$: **type**.
env : **type**.
$stack$: **type**.
$state$: **type**.

Unary numbers n

0 : $unary$.
$n+1$: $unary$.

Term t

x : $term$.
$t_1\,t_2$: $term$.
$\lambda x.t$: $term$.

Closure c

$(t,\mathcal{E})$: $clos$.

Environment $\mathcal{E}$

$[]$: env.
$(\mathcal{E},\ x \leftarrow c)$: env.

Stack $\mathcal{S}$

$[]$: $stack$.
$c : \mathcal{S}$: $stack$.

State σ

$\langle t,\mathcal{E},\mathcal{S}\rangle$: $state$.

1.2 Judgments

$\mathcal{E}(x) = c$: **type**.
$\sigma_1 \to \sigma_2$: **type**.

1.3 Fetch

$$\frac{}{(\mathcal{E},\ x \leftarrow c)(x) = c}\ [\text{Fetch}_1]$$

$$\frac{x \neq x' \quad \mathcal{E}(x) = c}{(\mathcal{E},\ x' \leftarrow c')(x) = c}\ [\text{Fetch}_2]$$

%mode $+\mathcal{E}(+x) = -c$
%worlds $()\quad \mathcal{E}(x) = c$
%terminates $\mathcal{E}\quad \mathcal{E}(x) = c$

Remark. Twelf cannot check the following property:
%unique $+\mathcal{E}(+x) = -1c$

1.4 Evaluation

$$\frac{\mathcal{E}(x) = (t, \mathcal{E}')}{\langle x, \mathcal{E}, \mathcal{S}\rangle \rightarrow \langle t, \mathcal{E}', \mathcal{S}\rangle}\text{[E\_Var]}$$

$$\frac{}{\langle (t_1\, t_2), \mathcal{E}, \mathcal{S}\rangle \rightarrow \langle t_1, \mathcal{E}, (t_2, \mathcal{E}) : \mathcal{S}\rangle}\text{[E\_App]}$$

$$\frac{}{\langle \lambda x.t, \mathcal{E}, c : \mathcal{S}\rangle \rightarrow \langle t, (\mathcal{E},\ x \leftarrow c), \mathcal{S}\rangle}\text{[E\_Abs]}$$

%mode $+\sigma_1 \rightarrow -\sigma_2$
%worlds $()\quad \sigma_1 \rightarrow \sigma_2$

Remark. Twelf cannot check the following property:
%unique $+\sigma_1 \rightarrow -1\sigma_2$