Syntax

$$
\begin{array}{ll}
e = x \mid \lambda x.e \mid e\,e & \text{(Expressions)} \\
c = e \mid c^{x} & \text{(Stepper Expressions)} \\
v = \lambda x.e & \text{(Values)} \\
a = A[v] & \text{(Answers)} \\
A = [\,] \mid A[\lambda x.A]\,e & \text{(Answer Contexts)} \\
\hat{A} = [\,] \mid A[\hat{A}]\,e & \text{(Answer Contexts – outer)} \\
\check{A} = [\,] \mid A[\lambda x.\check{A}] & \text{(Answer Contexts – inner)} \\
E = [\,] \mid E\,e \mid A[E] \mid \hat{A}[A[\lambda x.\check{A}[E[x]]]\,E] & \text{(Evaluation Contexts)} \\
\qquad \hat{A}[\check{A}] \in A & \\
C = [\,] \mid \lambda x.C \mid C\,e \mid e\,C & \text{(Contexts)} \\
F = \texttt{mt} \mid (\texttt{arg}\ e) \mid (\texttt{lam}\ x) \mid (\texttt{body}\ x\ [\![F,\ldots]\!]\ [\![F,\ldots]\!]) & \text{(Frames)} \\
S = \langle e,\ [\![F,\ldots]\!]\rangle & \text{(Machine States)}
\end{array}
$$

Machine Transitions

push-arg

$$\langle e_1\,e_2,\ [\![F,\ldots]\!]\rangle \longmapsto_{ck} \langle e_1,\ [\![(\texttt{arg}\ e_2),F,\ldots]\!]\rangle$$

push-lam

$$\langle \lambda x.e,\ [\![F,\ldots]\!]\rangle \longmapsto_{ck} \langle e,\ [\![(\texttt{lam}\ x),F,\ldots]\!]\rangle$$

$$\text{more args than } \lambda\text{s in } [\![F,\ldots]\!]$$

lookup-var

$$\langle x,\ [\![F_1,\ldots,(\texttt{lam}\ x),F_2,\ldots,(\texttt{arg}\ e),F_3,\ldots]\!]\rangle \longmapsto_{ck} \langle e,\ [\![(\texttt{body}\ x\ [\![F_1,\ldots]\!]\ [\![F_2,\ldots]\!]),F_3,\ldots]\!]\rangle$$

$$
\begin{array}{r}
(\texttt{lam}\ x) \notin F_1,\ldots \\
\phi_F([\![F_1,\ldots]\!]) \in \check{A}[E] \\
\phi_F([\![F_2,\ldots]\!]) \in A \\
\phi_F([\![F_3,\ldots]\!]) \in E[\hat{A}] \\
\hat{A}[\check{A}] \in A
\end{array}
$$

$\boldsymbol{\beta}_{\textbf{need}}$

$$\langle v,\ [\![F_1,\ldots,(\texttt{body}\ x\ [\![F_3,\ldots]\!]\ [\![F_4,\ldots]\!]),F_2,\ldots]\!]\rangle \longmapsto_{ck} \langle v,\ [\![F_3\{x:=v\},\ldots,F_1,\ldots,F_4,\ldots,F_2,\ldots]\!]\rangle$$

$$\phi_F([\![F_1,\ldots]\!]) \in A$$

$$
\begin{aligned}
\phi : S &\to e \\
\phi(\langle e,\ [\![F,\ldots]\!]\rangle) &= \phi_F([\![F,\ldots]\!])[e]
\end{aligned}
$$

$$
\begin{aligned}
\phi_F : [\![F,\ldots]\!] &\to E \\
\phi_F([\![F,\ldots]\!]) &= \phi'_F([\![F,\ldots]\!],\ [\,])
\end{aligned}
$$

$$
\begin{aligned}
\phi' : [\![F,\ldots]\!] \times C &\to E \\
\phi'_F([\![\texttt{mt}]\!],\ C) &= C \\
\phi'_F([\![(\texttt{lam}\ x), F,\ldots]\!],\ C) &= \phi'_F([\![F,\ldots]\!],\ \lambda x.C) \\
\phi'_F([\![(\texttt{arg}\ e), F,\ldots]\!],\ C) &= \phi'_F([\![F,\ldots]\!],\ C\,e) \\
\phi'_F([\![(\texttt{body}\ x\ [\![F_1,\ldots]\!]\ [\![F_2,\ldots]\!]), F,\ldots]\!],\ C) &= \phi'_F([\![F,\ldots]\!],\ \phi_F([\![F_2,\ldots]\!])[\lambda x.\phi_F([\![F_1,\ldots]\!])[x]]\,C)
\end{aligned}
$$

$$
\begin{aligned}
\psi : S &\to c \\
\psi(\langle e,\ [\![F,\ldots]\!]\rangle) &= \psi'([\![F,\ldots]\!],\ e)
\end{aligned}
$$

$$
\begin{aligned}
\psi' : [\![F,\ldots]\!] \times c &\to c \\
\psi'([\![\texttt{mt}]\!],\ c) &= c \\
\psi'([\![(\texttt{lam}\ x), F_1,\ldots,(\texttt{arg}\ c_1), F_2,\ldots]\!],\ c) &= \psi'([\![F_1,\ldots,F_2,\ldots]\!],\ c\{x := c_1^x\}) \\
\phi_F([\![F_1,\ldots]\!]) &\in A \\
\psi'([\![(\texttt{arg}\ c_1), F,\ldots]\!],\ c) &= \psi'([\![F,\ldots]\!],\ c\,c_1) \\
\psi'([\![(\texttt{body}\ x\ [\![F_1,\ldots]\!]\ [\![F_2,\ldots]\!]), F,\ldots]\!],\ c) &= \psi'([\![F_1,\ldots,(\texttt{lam}\ x), F_2,\ldots,(\texttt{arg}\ c), F,\ldots]\!],\ x)
\end{aligned}
$$

Theorem 1. *For a program e, if $\langle e,\ [\![\mathtt{mt}]\!]\rangle \longmapsto\!\!\!\!\rightarrow_{ck} \langle v,\ [\![F,\ldots]\!]\rangle$, then for all steps $S_1 \longmapsto_{ck} S_2$ in the sequence, either:*

1. $\phi(S_1) = \phi(S_2)$, *or*
2. $\phi(S_1) \longmapsto \phi(S_2)$

Proof. By cases on $S_1 \longmapsto_{ck} S_2$:

Case *push-arg***.**

first claim

Case *push-lam***.**

first claim

Case *lookup-var***.**

first claim

Case $\boldsymbol{\beta}_{\mathbf{need}}$**.**

second claim

□

Theorem 2. *For a program e, if $\langle e,\ [\![\mathtt{mt}]\!]\rangle \longmapsto\!\!\!\!\rightarrow_{ck} \langle v,\ [\![F,\ldots]\!]\rangle$, then for all steps $S_1 \longmapsto_{ck} S_2$ in the sequence, either:*

1. $\psi(S_1) = \psi(S_2)$, *or*
2. $\psi(S_1) \longmapsto_{stepper} \psi(S_2)$

Proof. By cases on $S_1 \longmapsto_{ck} S_2$:

Case *push-arg***.**

first claim

Case *push-lam***.**

second claim

Case *lookup-var***.**

first claim

Case $\boldsymbol{\beta}_{\mathbf{need}}$**.**

first claim

□