ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC

ROSAEC MEMO
2012-015
March 15, 2012

Unstaging Translation from MetaML-Like Multi-Staged Calculus to Context Calculus

Joonwon Choi
jwchoi@ropas.snu.ac.kr

Jeehoon Kang
jhkang@ropas.snu.ac.kr

Daejun Park
djpark@ropas.snu.ac.kr

Kwangkeun Yi
kwang@ropas.snu.ac.kr

March 15, 2012

1 MetaML-Like Multi-Staged Calculus $\lambda_{\mathcal{S}}$

1.1 Syntax

The syntax of MetaML-like multi-staged calculus is defined as Figure 1.

MetaML-like multi-staged calculus admits `lift` as follows:

$$\texttt{lift}\ e =_{\texttt{def}} \lambda x.\texttt{box}\ x\ e$$

It does not hold in Lisp-like multi-staged calculus [1] because the defined x and the used x belong to different stages. Note that `lift` is not admissible in Lisp-like multi-staged calculus.

Syntax

Variable	x, y	$\in$	*Var*
Constant	i	$\in$	*Const*
$\mathit{Expr}_{\mathcal{S}}$	e	$::=$	$i \mid x \mid \lambda x.e \mid e\ e$
		$\mid$	$\texttt{box}\ e \mid \texttt{unbox}\ e \mid \texttt{run}\ e$

Figure 1: Syntax of $\lambda_{\mathcal{S}}$

1.2 Operational Semantics

The operational semantics of $\lambda_{\mathcal{S}}$ and the substitution rule for that is given as Figure 2 and Figure 3, respectively.

Multi-staged calculus has many syntactic categories for values and expressions including n-staged values, $\mathit{Value}^n_{\mathcal{S}}$, and n-staged expressions, $\mathit{Expr}^n_{\mathcal{S}}$. The operational semantics depends on the diversified syntactic categories to rule out ill-formed programs.

The substitution in MetaML-like multi-staged calculus works across staging constructs.

To clarify n-staged values and expressions, we define the *depth* of expressions. For $e \in \mathit{Expr}_{\mathcal{S}}$, depth of e, $\mathit{depth}(e)$, is the number of nested `unbox`es that are not enclosed with `box`es, defined as Definition 1.

Definitions

$$
\begin{array}{ll}
Value^0_{\mathcal{S}} & v^0 ::= i \mid x \mid \lambda x.e^0 \mid \texttt{box}\ v^1 \\
Value^n_{\mathcal{S}}(n \geq 1) & v^n ::= i \mid x \mid \lambda x.v^n \mid v^n\ v^n \\
 & \quad \mid \texttt{box}\ v^{n+1} \mid \texttt{unbox}\ v^{n-1} (n \geq 2) \mid \texttt{run}\ v^n \\
Expr^0_{\mathcal{S}} & e^0 ::= i \mid x \mid \lambda x.e^0 \mid e^0\ e^0 \mid \texttt{box}\ e^1 \mid \texttt{run}\ e^0 \\
Expr^n_{\mathcal{S}}(n \geq 1) & e^n ::= i \mid x \mid \lambda x.e^n \mid e^n\ e^n \\
 & \quad \mid \texttt{box}\ e^{n+1} \mid \texttt{unbox}\ e^{n-1} (n \geq 1) \mid \texttt{run}\ e^n
\end{array}
$$

Operational Semantics

$$
(\texttt{APP}_{\mathcal{S}}) \quad \frac{e_1 \xrightarrow{n} e_1'}{e_1\ e_2^n \xrightarrow{n} e_1'\ e_2^n} \qquad \frac{e \xrightarrow{n} e'}{v^n\ e \xrightarrow{n} v^n\ e'} \qquad \frac{}{(\lambda x.e^0)\ v^0 \xrightarrow{0} [x \mapsto v^0]e^0}
$$

$$
(\texttt{BOX}_{\mathcal{S}}) \quad \frac{e \xrightarrow{n+1} e'}{\texttt{box}\ e \xrightarrow{n} \texttt{box}\ e'}
$$

$$
(\texttt{UNB}_{\mathcal{S}}) \quad \frac{e \xrightarrow{n} e'}{\texttt{unbox}\ e \xrightarrow{n+1} \texttt{unbox}\ e'} \qquad \frac{}{\texttt{unbox}\ \left(\texttt{box}\ v^1\right) \xrightarrow{1} v^1}
$$

$$
(\texttt{RUN}_{\mathcal{S}}) \quad \frac{e \xrightarrow{n} e'}{\texttt{run}\ e \xrightarrow{n} \texttt{run}\ e'} \qquad \frac{}{\texttt{run}\ \left(\texttt{box}\ v^1\right) \xrightarrow{0} v^1}
$$

$$
(\texttt{ABS}_{\mathcal{S}}) \quad \frac{e \xrightarrow{n+1} e'}{\lambda x.e \xrightarrow{n+1} \lambda x.e'}
$$

Figure 2: Operational Semantics of $\lambda_{\mathcal{S}}$

Substitution

$$
\begin{array}{lll}
[x \mapsto v]i & = i & \\
[x \mapsto v]y & = \begin{cases} v & \text{if } x = y \\ y & \text{otherwise} \end{cases} & \\
[x \mapsto v]\lambda y.e & = \begin{cases} \lambda y.e & \text{if } x = y \text{ or } x \notin FV(e) \\ \lambda y.[x \mapsto v]e & \text{if } x \neq y \text{ and } y \notin FV(v) \text{ and } y \notin BV(v) \\ \lambda y'.[x \mapsto v]([y \mapsto y']e) & \text{otherwise, where } y' \notin FV(v) \text{ and } y' \notin BV(v) \end{cases} & \\
[x \mapsto v](e_1\ e_2) & = [x \mapsto v]e_1\ [x \mapsto v]e_2 & \\
 & & \\
[x \mapsto v]\texttt{box}\ e & = \texttt{box}\ [x \mapsto v]e & \\
[x \mapsto v]\texttt{unbox}\ e & = \texttt{unbox}\ [x \mapsto v]e & \\
[x \mapsto v]\texttt{run}\ e & = \texttt{run}\ [x \mapsto v]e &
\end{array}
$$

Free/Bound Variables

$$
\begin{array}{llll}
FV(i) & = \varnothing & BV(i) & = \varnothing \\
FV(x) & = \{x\} & BV(x) & = \varnothing \\
FV(\lambda x.e) & = FV(e) \setminus \{x\} & BV(\lambda x.e) & = BV(e) \cup \{x\} \\
FV(e_1\ e_2) & = FV(e_1) \cup FV(e_2) & BV(e_1\ e_2) & = BV(e_1) \cup BV(e_2) \\
FV(\texttt{box}\ e) & = FV(e) & BV(\texttt{box}\ e) & = BV(e) \\
FV(\texttt{unbox}\ e) & = FV(e) & BV(\texttt{unbox}\ e) & = BV(e) \\
FV(\texttt{run}\ e) & = FV(e) & BV(\texttt{run}\ e) & = BV(e)
\end{array}
$$

Figure 3: Substitution over $\lambda_{\mathcal{S}}$

Definition 1 (Depth)**.** *For $e \in Expr_{\mathcal{S}}$, $depth(e)$ is defined as the following.*

$$
\begin{array}{ll}
depth(i) & = 0 \\
depth(x) & = 0 \\
depth(\lambda x.e) & = depth(e) \\
depth(e_1\ e_2) & = max(depth(e_1), depth(e_2)) \\
depth(\texttt{box}\ e) & = \begin{cases} depth(e) - 1 & \text{if } depth(e) > 0 \\ 0 & \text{if } depth(e) = 0 \end{cases} \\
depth(\texttt{unbox}\ e) & = depth(e) + 1 \\
depth(\texttt{run}\ e) & = depth(e)
\end{array}
$$

Clearly, $depth(e) \geq 0$.

Intuitively, the depth means the minimum stage of which the given expression is valid. For example, if $depth(e) = n$, e is meaningless in semantics at the stage below n.

Furthermore, the depth of e determines the reducibility of e at the stage n. Suppose an expression e is not stuck and the depth of e is equal to or less than $n > 0$. The expressions e is reducible at stage n if and only if the depth of e is n. Lemma 1 states these properties. Refer to § 5.2 for the proof of Lemma 1.

Lemma 1 (Reducibility wrt Depth)**.** *Depth determines the reducibility as the follows:*

- *If $e \in Expr^n_{\mathcal{S}}$ then $depth(e) \leq n$.*
- *If $v \in Value^n_{\mathcal{S}}$ then* $\begin{cases} depth(v) = 0 & \text{if } n = 0 \\ depth(v) < n & \text{if } n \geq 1 \end{cases}$
- *If $e \xrightarrow{n} e'$ then $depth(e) = n$.*

2 Context Calculus $\lambda_{\mathcal{C}}$

2.1 Syntax

The syntax of $\lambda_{\mathcal{C}}$ [2] is defined as Figure 4. Note that the type system of $\lambda_{\mathcal{C}}$ [2] tells us that for any hole abstraction $\delta X.e$ the hole variable X occurs only once at e and does not occur outside the abstraction. Here, we do not introduce the type system of $\lambda_{\mathcal{C}}$ explicitly.

The renamer ν is a partial function from *Var* to *Var*, denoted by $\{y_1/x_1, \cdots, y_n/x_n\}$, where y_i's are fresh names without conflicting x_i's for the simplicity, (except identity renamers). The renamer can be easily extended into a total function by letting $\nu(x) = x$ for all $x \notin dom(\nu)$. Therefore, we can use $\{\}$ for an identity function.

Syntax

Variable	$x, y, u, h, w_0, w_1, \cdots$	$\in$	Var
Hole Variable	X, Y, H	$\in$	$HoleVar$
Renamer	ν	$\in$	$Var \xrightarrow{\texttt{fin}} Var$
Constant	i	$\in$	$Const$
$Expr_{\mathcal{C}}$	e	$::=$	$i \mid x \mid \lambda x.e \mid e\ e$
		$\mid$	$X^{\nu} \mid \delta X.e \mid e \odot_{\nu} e$

Figure 4: Syntax of $\lambda_{\mathcal{C}}$

2.2 Operational Semantics

The operational semantics of $\lambda_{\mathcal{C}}$ and the substitution rule for that is given as Figure 5 and Figure 6, respectively.

Definitions

$$Value_{\mathcal{C}} \quad v ::= i \mid x \mid X \mid \lambda x.e \mid \delta X.e$$

Operational Semantics

$$(\texttt{APP}_{\mathcal{C}}) \quad \frac{e_1 \xrightarrow{\mathcal{C}} e_1'}{e_1\ e_2 \xrightarrow{\mathcal{C}} e_1'\ e_2} \qquad \frac{e \xrightarrow{\mathcal{C}} e'}{v\ e \xrightarrow{\mathcal{C}} v\ e'} \qquad \frac{FH(e) = FH(v) = \varnothing}{(\lambda x.e)\ v \xrightarrow{\mathcal{C}} \{v/x\}e}$$

$$(\texttt{HAPP}_{\mathcal{C}}) \quad \frac{e_1 \xrightarrow{\mathcal{C}} e_1'}{e_1 \odot_{\nu} e_2 \xrightarrow{\mathcal{C}} e_1' \odot_{\nu} e_2} \qquad \frac{e \xrightarrow{\mathcal{C}} e'}{v \odot_{\nu} e \xrightarrow{\mathcal{C}} v \odot_{\nu} e'} \qquad \frac{}{(\delta X.e) \odot_{\nu} v \xrightarrow{\mathcal{C}} (e[X^{\nu}/X])[v/X]}$$

Figure 5: Operational Semantics of $\lambda_{\mathcal{C}}$

Variable Substitution

$$
\begin{aligned}
\{v/x\}i &= i \\
\{v/x\}y &= \begin{cases} v & \text{if } x = y \\ y & \text{otherwise} \end{cases} \\
\{v/x\}\lambda y.e &= \begin{cases} \lambda y.e & \text{if } x = y \text{ or } x \notin FV(e) \\ \lambda y.\{v/x\}e & \text{if } x \neq y \text{ and } y \notin FV(v) \text{ and } y \notin BV(v) \\ \lambda y'.\{v/x\}(\overline{\{y'/y\}}\ e) & \text{otherwise, where } y' \notin FV(v) \text{ and } y' \notin BV(v) \end{cases} \\
\{v/x\}(e_1\ e_2) &= \{v/x\}e_1\ \{v/x\}e_2 \\
\\
\{v/x\}X^{\nu} &= X^{\nu} \\
\{v/x\}\delta X.e &= \delta X.\{v/x\}e \\
\{v/x\}e_1 \odot_{\nu} e_2 &= \begin{cases} \{v/x\}e_1 \odot_{\nu} \{v/x\}e_2 & \text{if } x \notin dom(\nu) \\ \{v/x\}e_1 \odot_{\nu} e_2 & \text{if } x \in dom(\nu) \end{cases}
\end{aligned}
$$

Hole Substitution

$$
\begin{aligned}
i[v/X] &= i \\
x[v/X] &= x \\
(\lambda x.e)[v/X] &= \lambda x.e[v/X] \\
(e_1\ e_2)[v/X] &= e_1[v/X]\ e_2[v/X]
\end{aligned}
\qquad
\begin{aligned}
Y^{\nu}[v/X] &= \begin{cases} \overline{\nu}\ v & \text{if } X = Y \\ Y^{\nu} & \text{if } X \neq Y \end{cases} \\
(\delta Y.e)[v/X] &= \begin{cases} \delta Y.e & \text{if } X = Y \\ \delta Y.e[v/X] & \text{if } X \neq Y \text{ and } Y \notin FH(v) \end{cases} \\
(e_1 \odot_{\nu} e_2)[v/X] &= e_1[v/X] \odot_{\nu} e_2[v/X]
\end{aligned}
$$

Free/Bound Variables

$$
\begin{aligned}
FV(i) &= \varnothing \\
FV(x) &= \{x\} \\
FV(\lambda x.e) &= FV(e) \setminus \{x\} \\
FV(e_1\ e_2) &= FV(e_1) \cup FV(e_2) \\
FV(X^{\nu}) &= \varnothing \\
FV(\delta X.e) &= FV(e) \\
FV(e_1 \odot_{\nu} e_2) &= FV(e_1) \cup (FV(e_2) \setminus dom(\nu))
\end{aligned}
\qquad
\begin{aligned}
BV(i) &= \varnothing \\
BV(x) &= \varnothing \\
BV(\lambda x.e) &= BV(e) \cup \{x\} \\
BV(e_1\ e_2) &= BV(e_1) \cup BV(e_2) \\
BV(X^{\nu}) &= \varnothing \\
BV(\delta X.e) &= BV(e) \\
BV(e_1 \odot_{\nu} e_2) &= BV(e_1) \cup (BV(e_2) \cup dom(\nu))
\end{aligned}
$$

Free/Bound Hole Variables

$$
\begin{aligned}
FH(i) &= \varnothing \\
FH(x) &= \varnothing \\
FH(\lambda x.e) &= FH(e) \\
FH(e_1\ e_2) &= FH(e_1) \cup FH(e_2) \\
FH(X^{\nu}) &= \{X\} \\
FH(\delta X.e) &= FH(e) \setminus \{X\} \\
FH(\delta e_1.e_2 \nu) &= FH(e_1) \cup FH(e_2)
\end{aligned}
\qquad
\begin{aligned}
BH(i) &= \varnothing \\
BH(x) &= \varnothing \\
BH(\lambda x.e) &= BH(e) \\
BH(e_1\ e_2) &= BH(e_1) \cup BH(e_2) \\
BH(X^{\nu}) &= \{X\} \\
BH(\delta X.e) &= BH(e) \cup \{X\} \\
BH(\delta e_1.e_2 \nu) &= BH(e_1) \cup BH(e_2)
\end{aligned}
$$

Renamer Application

$$
\begin{aligned}
\overline{\nu}\ x &= \nu(x) \\
\overline{\nu}\ \lambda x.e &= \lambda x.\overline{\nu|_{dom(\nu)\setminus\{x\}}}\ e \\
\overline{\nu}\ e_1\ e_2 &= \overline{\nu}\ e_1\ \overline{\nu}\ e_2
\end{aligned}
\qquad
\begin{aligned}
\overline{\nu}\ X^{\nu'} &= X^{\nu \star \nu'} \\
\overline{\nu}\ \delta X.e &= \delta X.\overline{\nu}\ e \\
\overline{\nu}\ e_1 \odot_{\nu'} e_2 &= \overline{\nu}\ e_1 \odot_{\nu'} \overline{\nu|_{dom(\nu)\setminus dom(\nu')}}\ e_2
\end{aligned}
$$

Renamer Composition

$$
\nu_1 \star \nu_2 = \begin{cases} \nu_1 & \text{if } \nu_2 = \{\} \\ \nu_2 & \text{if } \nu_1 = \{\} \\ \nu & \text{o.w., where } dom(\nu) = dom(\nu_2) \text{ and } \forall x \in dom(\nu_2).\ \nu(x) = \nu_1(\nu_2(x)) \end{cases}
$$

Figure 6: Substitution over $\lambda_{\mathcal{C}}$

3 Translation and Simulation

3.1 Inference Rules

Definitions

Environment	$r ::= \bot \mid r;x$
Environment Stack	$R ::= r \mid R,r$
Continuation	$\kappa ::= (\delta H.[\cdot]) \odot_\nu e \mid (\delta H.\kappa) \odot_\nu e$
Continuation Stack	$K ::= \bot \mid K,\kappa$

Translation

(TCON) $R \vdash i \mapsto (i, \bot)$ (TVAR) $R \vdash x \mapsto (x, \bot)$

(TABS) $$\frac{R,(r_n;x) \vdash e \mapsto (\underline{e}, K)}{R, r_n \vdash \lambda x.e \mapsto (\lambda x.\underline{e}, K)}$$

(TAPP) $$\frac{R \vdash e_1 \mapsto (\underline{e_1}, K_1) \qquad R \vdash e_2 \mapsto (\underline{e_2}, K_2)}{R \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2)}$$

(TBOX) $$\frac{R, \bot \vdash e \mapsto (\underline{e}, (K,\kappa)) \qquad \texttt{new}\ u}{R \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], K)} \qquad \frac{R, \bot \vdash e \mapsto (\underline{e}, \bot) \qquad \texttt{new}\ u}{R \vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot)}$$

(TUNB) $$\frac{\begin{array}{c} R, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, K) \\ r_n = x_k; \cdots; x_0 \qquad \nu = \{w_k/x_k, \cdots, w_0/x_0\} \qquad \texttt{new}\ H \end{array}}{R, r_{n-1}, r_n \vdash \texttt{unbox}\ e \mapsto (H^{\nu^{-1}}\ \texttt{()}, (K, ((\delta H.[\cdot]) \odot_\nu \underline{e})))}$$

(TRUN) $$\frac{R \vdash e \mapsto (\underline{e}, K) \qquad \texttt{new}\ h}{R \vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ h\ \texttt{()}, K)}$$

Environment Concatenation

$$\begin{array}{rcl} r;\bot & = & r \\ \bot;r & = & r \\ r_1;(r_2;x) & = & (r_1;r_2);x \end{array}$$

Renamer Inverse

$$\{y_1/x_1, \cdots, y_n/x_n\}^{-1} = \{x_1/y_1, \cdots, x_n/y_n\}$$

Continuation Stack Merger

$$\begin{array}{rcl} \bot \bowtie K & = & K \\ K \bowtie \bot & = & K \\ (K_1,\kappa_1) \bowtie (K_2,\kappa_2) & = & (K_1 \bowtie K_2), (\kappa_1[\kappa_2]) \end{array}$$

Figure 7: Translation from $\lambda_{\mathcal{S}}$ to $\lambda_{\mathcal{C}}$

3.2 Simulation

We need to introduce the admin reduction as Figure 8.

We state the correctness of the translation in terms of the simulation of reducible expressions and values of $\lambda_{\mathcal{S}}$. Note that we do not need to assume that $FV(e) = \varnothing$. Our translation works with any expression with free variables. Refer to § 5.6 for the proof of Theorem 1.

Theorem 1 (Simulation)**.** *Let* $e, e' \in Expr_{\mathcal{S}}$ *and* $e \xrightarrow{0} e'$. *Let* $\bot \vdash e \mapsto (\underline{e}, \bot)$ *and* $\bot \vdash e' \mapsto (\underline{e'}, \bot)$. *Then* $\underline{e} \xrightarrow{\mathcal{C};\mathcal{A}^*} \underline{e'}$. *Furthermore, If* $e \in Value^0_{\mathcal{S}}$ *then* $\underline{e} \in Value_{\mathcal{C}}$.

Admin Reduction

$$(\texttt{ABS}_{\mathcal{A}}) \qquad (\lambda u.e)\,() \xrightarrow{\mathcal{A}} \{()/u\}e$$

$$(\texttt{ABS}_{\mathcal{A}}) \quad \frac{e \xrightarrow{\mathcal{A}} e'}{\lambda x.e \xrightarrow{\mathcal{A}} \lambda x.e'} \qquad (\texttt{APP}_{\mathcal{A}}) \quad \frac{e_1 \xrightarrow{\mathcal{A}} e_1'}{e_1\ e_2 \xrightarrow{\mathcal{A}} e_1'\ e_2} \qquad \frac{e_2 \xrightarrow{\mathcal{A}} e_2'}{e_1\ e_2 \xrightarrow{\mathcal{A}} e_1\ e_2'}$$

$$(\texttt{HABS}_{\mathcal{A}}) \quad \frac{e \xrightarrow{\mathcal{A}} e'}{\delta X.e \xrightarrow{\mathcal{A}} \delta X.e'} \qquad (\texttt{HAPP}_{\mathcal{A}}) \quad \frac{e_1 \xrightarrow{\mathcal{A}} e_1'}{e_1 \odot_\nu e_2 \xrightarrow{\mathcal{A}} e_1' \odot_\nu e_2} \qquad \frac{e_2 \xrightarrow{\mathcal{A}} e_2'}{e_1 \odot_\nu e_2 \xrightarrow{\mathcal{A}} e_1 \odot_\nu e_2'}$$

Figure 8: Admin Reduction

4 Inverse Translation

4.1 Inference Rules

Theorem 2 (Inversion)**.** *Let e be a $\lambda_{\mathcal{S}}$ expression and R be an environment stack. If $R \vdash e \mapsto (\underline{e}, K)$ then $\mathcal{H} \vdash \underline{e} \rightarrowtail e$ for any $\mathcal{H}$ such that $\overline{K} \subseteq \mathcal{H}$.*

Definitions

$$\textit{Hole Environment} \quad \mathcal{H} \in \textit{HoleVar} \xrightarrow{\texttt{fin}} \textit{Expr}_{\mathcal{C}}$$

Translation

(ICON) $$\mathcal{H} \vdash i \rightarrowtail i$$

(IVAR) $$\mathcal{H} \vdash x \rightarrowtail x$$

(IABS) $$\frac{\mathcal{H} \vdash \underline{e} \rightarrowtail e}{\mathcal{H} \vdash \lambda x.\underline{e} \rightarrowtail \lambda x.e}$$

(IAPP) $$\frac{\mathcal{H} \vdash \underline{e_i} \rightarrowtail e_i \quad \underline{e_2} \neq \texttt{()}}{\mathcal{H} \vdash \underline{e_1}\ \underline{e_2} \rightarrowtail e_1\ e_2}$$

(ICTX) $$\frac{\mathcal{H} \cup \{H : \underline{e'}\} \vdash \underline{e} \rightarrowtail e}{\mathcal{H} \vdash (\delta H.\underline{e}) \odot_\nu \underline{e'} \rightarrowtail e}$$

(IBOX) $$\frac{\mathcal{H} \vdash \underline{e} \rightarrowtail e}{\mathcal{H} \vdash \lambda u.\underline{e} \rightarrowtail \texttt{box}\ e}$$

(IUNB) $$\frac{\mathcal{H} \vdash \mathcal{H}(H) \rightarrowtail e}{\mathcal{H} \vdash H\ \texttt{()} \rightarrowtail \texttt{unbox}\ e}$$

(IRUN) $$\frac{\mathcal{H} \vdash \underline{e} \rightarrowtail e}{\mathcal{H} \vdash \texttt{let}\ H = \underline{e}\ \texttt{in}\ (H\ \texttt{()}) \rightarrowtail \texttt{run}\ e}$$

Continuation Cumulation Operator

$$\begin{aligned}
\overline{\bot} &= \varnothing \\
\overline{(K,\kappa)} &= \overline{K} \cup \overline{\kappa} \\
\overline{[\cdot]} &= \varnothing \\
\overline{(\delta H.\kappa) \odot_\nu \underline{e}} &= \overline{\kappa} \cup \{H : \underline{e}\}
\end{aligned}$$

Figure 9: Inverse Translation from $\lambda_{\mathcal{C}}$ to $\lambda_{\mathcal{S}}$

5 Proofs

5.1 Preliminaries

5.1.1 Notations

Notation 1 (Abbreviations)**.** *If K is not $\bot$, we omit the leading $\bot$ from K. That is, we use $(\kappa_1, \cdots, \kappa_n)$ instead of $(\bot, \kappa_1, \cdots, \kappa_n)$.*

Notation 2. *We write $\mathcal{L}(Var)$ for a set of lists consisting of elements of Var. For $x \in Var$ and $r \in \mathcal{L}(Var)$, we write $x \in r$ meaning that x is an element of r, and we write $x \notin r$ meaning that x is not an element of r. We write $r \setminus \{x\}$ for the list that is same with r but does not the element x.*

5.1.2 Assumptions

To avoid unnecessary complexities, we assume that for all expression e of $\lambda_{\mathcal{S}}$, all bound variables in e are distinct. This results in no loss of generality.

5.2 Continuations

We need to extend some definitions of $\lambda_{\mathcal{C}}$ into the continuations and the continuation stacks for clarity of proofs.

The following remarks are corollary of Figure 10.

Remark 1. Let $x \in Var$, $X \in HoleVar$, $v \in Value_{\mathcal{C}}$ and $K \in$ *Continuation Stack*. Then

- $\{v/x\}(K_1 \bowtie K_2) = \{v/x\}K_1 \bowtie \{v/x\}K_2$
- $(K_1 \bowtie K_2)[v/X] = K_1[v/X] \bowtie K_2[v/X]$

We will start with the proof of Lemma 1.

Proof of Lemma 1. This proof is divided into two parts. The first is for the first two statememts (S1) and (S2), and the second is for the last statement (S3).

`Part 1:  Proof of (S1) and (S2)`

By structural induction on $Expr_{\mathcal{S}}$ (IH1) and $Value_{\mathcal{S}}$ (IH2), and case analysis on rules.

- Case $e^0 \in Expr^0_{\mathcal{S}}$.
  - Case $e^0 = i$. $depth(i) = 0 \le 0$.
  - Case $e^0 = x$. $depth(x) = 0 \le 0$.
  - Case $e^0 = \lambda x.e^0$. By (IH1), $depth(e^0) \le 0$. Then, $depth(\lambda x.e^0) = depth(e^0) \le 0$.
  - Case $e^0 = \texttt{box}\ e^1$. By (IH1), $depth(e^1) \le 1$. Then, $depth(e^1)$ is 0 or 1.
    - Case $depth(e^1) = 0$. $depth(\texttt{box}\ e^1) = 0 \le 0$.
    - Case $depth(e^1) = 1$. $depth(\texttt{box}\ e^1) = depth(e^1) - 1 = 1 - 1 = 0 \le 0$.
- Case $e^n \in Expr^n_{\mathcal{S}}$ $(n \ge 1)$.
  - Case $e^n = i$. $depth(i) = 0 \le n$.
  - Case $e^n = x$. $depth(x) = 0 \le n$.
  - Case $e^n = \lambda x.e^n$. By (IH1), $depth(e^n) \le n$. Then, $depth(\lambda x.e^n) = depth(e^n) \le n$.
  - Case $e^n = e^n_1\ e^n_2$. By (IH1), $depth(e^n_1) \le n$. By (IH1), $depth(e^n_2) \le n$. Then, $depth(e^n_1\ e^n_2) = max(depth(e^n_1), depth(e^n_2)) \le n$.

Definitions

$$x \in \mathit{Var} \qquad v \in \mathit{Value}_{\mathcal{C}} \qquad \kappa \in \mathit{Continuation}$$
$$X, H \in \mathit{HoleVar} \qquad e \in \mathit{Expr}_{\mathcal{C}} \qquad K \in \mathit{Continuation\ Stack}$$

Variable Substitution Extended into Continuations

$$\begin{aligned}
\{v/x\}[\cdot] &= [\cdot] \\
\{v/x\}(\delta H.\kappa) \odot_\nu e &= \begin{cases} (\delta H.\{v/x\}\kappa) \odot_\nu \{v/x\}e & \text{if } x \notin dom(\nu) \\ (\delta H.\{v/x\}\kappa) \odot_\nu e & \text{if } x \in dom(\nu) \end{cases} \\
\{v/x\}\bot &= \bot \\
\{v/x\}(K, \kappa) &= \{v/x\}K, \{v/x\}\kappa
\end{aligned}$$

Hole Substitution Extended into Continuations

$$\begin{aligned}
[\cdot][v/X] &= [\cdot] \\
(\delta H.\kappa \odot_\nu e)[v/X] &= \begin{cases} (\delta H.\kappa) \odot_\nu e[v/X] & \text{if } X = H \\ (\delta H.\kappa[v/X]) \odot_\nu e[v/X] & \text{if } X \neq H \end{cases} \\
\bot[v/X] &= \bot \\
(K, \kappa)[v/X] &= K[v/X], \kappa[v/X]
\end{aligned}$$

Admin Reduction Extended into Continuations

$$\frac{e \xrightarrow{\mathcal{A}} e'}{(\delta H.[\cdot]) \odot_\nu e \xrightarrow{\mathcal{A}} (\delta H.[\cdot]) \odot_\nu e'}$$

$$\frac{e \xrightarrow{\mathcal{A}} e'}{(\delta H.\kappa) \odot_\nu e \xrightarrow{\mathcal{A}} (\delta H.\kappa) \odot_\nu e'} \qquad \frac{\kappa \xrightarrow{\mathcal{A}} \kappa'}{(\delta H.\kappa) \odot_\nu e \xrightarrow{\mathcal{A}} (\delta H.\kappa') \odot_\nu e}$$

$$\frac{K \xrightarrow{\mathcal{A}} K'}{K, \kappa \xrightarrow{\mathcal{A}} K', \kappa} \qquad \frac{\kappa \xrightarrow{\mathcal{A}} \kappa'}{K, \kappa \xrightarrow{\mathcal{A}} K, \kappa'}$$

Figure 10: Definitions Extended into Continuations

- Case $e^n = \texttt{box}\ e^{n+1}$. By (IH1), $depth(e^{n+1}) \le n+1$. Then, $depth(\texttt{box}\ e^{n+1}) = depth(e^{n+1}) - 1 \le n+1-1 = n$.
- Case $e^n = \texttt{unbox}\ e^{n-1}$ $(n \ge 2)$. By (IH1), $depth(e^{n-1}) \le n-1$. Then, $depth(\texttt{unbox}\ e^{n-1}) = depth(e^{n-1}) + 1 \le n-1+1 = n$.
- Case $e^n = \texttt{run}\ e^n$. By (IH1), $depth(e^n) \le n$. Then, $depth(\texttt{run}\ e^n) = depth(e^n) \le n$.

- Case $v^0 \in Value^0_{\mathcal{S}}$.
  - Case $v^0 = i$. $depth(i) = 0$.
  - Case $v^0 = \lambda x.e^0$. By (IH1), $depth(e^0) \le 0$. Then, $depth(e^0) = 0$. Then, $depth(\lambda x.e^0) = depth(e^0) = 0$.
  - Case $v^0 = \texttt{box}\ v^1$. By (IH2), $depth(v^1) < 1$. Then, $depth(v^1) = 0$. Then, $depth(\texttt{box}\ v^1) = 0$.

- Case $v^n \in Value^n_{\mathcal{S}}$ $(n \ge 1)$.
  - Case $v^n = i$. $depth(i) = 0 < n$.
  - Case $v^n = x$. $depth(x) = 0 < n$.
  - Case $v^n = \lambda x.v^n$. By (IH2), $depth(v^n) < n$. Then, $depth(\lambda x.v^n) = depth(v^n) < n$.
  - Case $v^n = v_1^n\ v_2^n$. By (IH2), $depth(v_1^n) < n$. By (IH2), $depth(v_2^n) < n$. Then, $depth(v_1^n\ v_2^n) = max(depth(v_1^n), depth(v_2^n)) < n$.
  - Case $v^n = \texttt{box}\ v^{n+1}$. By (IH2), $depth(v^{n+1}) < n+1$. Then, $depth(\texttt{box}\ v^{n+1}) = depth(v^{n+1}) - 1 < n+1-1 = n$.
  - Case $v^n = \texttt{unbox}\ v^{n-1}$ $(n \ge 2)$. By (IH2), $depth(v^{n-1}) < n-1$. Then, $depth(\texttt{unbox}\ v^{n-1}) = depth(v^{n-1}) + 1 < n-1+1 = n$.
  - Case $v^n = \texttt{run}\ v^n$. By (IH2), $depth(v^n) < n$. Then, $depth(\texttt{run}\ v^n) = depth(v^n) < n$.

`Part 2: Proof of (S3)`

By structural induction on the derivation of $\cdot \xrightarrow{n} \cdot$ (IH).

- Case $(\texttt{APP}_{\mathcal{S}})_1$. $\dfrac{e_1 \xrightarrow{n} e_1'}{e_1\ e_2^n \xrightarrow{n} e_1'\ e_2^n}$. By (IH), $depth(e_1) = n$. By (S1), $depth(e_2^n) \le n$. Then, $depth(e_1\ e_2^n) = max(depth(e_1), depth(e_2^n)) = n$.
- Case $(\texttt{APP}_{\mathcal{S}})_2$.
  - $n = 0$. $\dfrac{e \xrightarrow{0} e'}{v^0\ e \xrightarrow{0} v^0\ e'}$. By (IH), $depth(e) = 0$. By (S2), $depth(v^0) = 0$. Then, $depth(v^0\ e) = max(depth(v^0), depth(e)) = 0$.
  - $n \ge 1$. $\dfrac{e \xrightarrow{n} e'}{v^n\ e \xrightarrow{n} v^n\ e'}$. By (IH), $depth(e) = n$. By (S2), $depth(v^n) \le n$. Then, $depth(v^n\ e) = max(depth(v^n), depth(e)) = n$.
- Case $(\texttt{APP}_{\mathcal{S}})_3$. $(\lambda x.e^0)\ v^0 \xrightarrow{0} [x \mapsto v^0]e^0$. By (S2), $depth(\lambda x.e^0) = 0$. By (S2), $depth(v^0) = 0$. Then, $depth((\lambda x.e^0)\ v^0) = max(depth(\lambda x.e^0), depth(v^0)) = 0$.
- Case $(\texttt{BOX}_{\mathcal{S}})$. $\dfrac{e \xrightarrow{n+1} e'}{\texttt{box}\ e \xrightarrow{n} \texttt{box}\ e'}$ By (IH), $depth(e) = n+1$. Then, $depth(\texttt{box}\ e) = depth(e) - 1 = n+1-1 = n$.

- Case $(\texttt{UNB}_{\mathcal{S}})_1$. $\dfrac{e \xrightarrow{n} e'}{\texttt{unbox}\ e \xrightarrow{n+1} \texttt{unbox}\ e'}$ By (IH), $depth(e) = n$. Then, $depth(\texttt{unbox}\ e) = depth(e) + 1 = n + 1$.
- Case $(\texttt{UNB}_{\mathcal{S}})_2$. $\dfrac{}{\texttt{unbox}\ \left(\texttt{box}\ v^1\right) \xrightarrow{1} v^1}$ By (S2), $depth(\texttt{box}\ v^1) = 0$. Then, $depth(\texttt{unbox}\ \left(\texttt{box}\ v^1\right)) = depth(\texttt{box}\ v^1) + 1 = 1$.
- Case $(\texttt{RUN}_{\mathcal{S}})_1$. $\dfrac{e \xrightarrow{n} e'}{\texttt{run}\ e \xrightarrow{n} \texttt{run}\ e'}$ By (IH), $depth(e) = n$. Then, $depth(\texttt{run}\ e) = depth(e) = n$.
- Case $(\texttt{RUN}_{\mathcal{S}})_2$. $\dfrac{FV(v^1) = \varnothing}{\texttt{run}\ \left(\texttt{box}\ v^1\right) \xrightarrow{0} v^1}$ By (S2), $depth(\texttt{box}\ v^1) = 0$. Then, $depth(\texttt{run}\ \left(\texttt{box}\ v^1\right)) = depth(\texttt{box}\ v^1) = 0$.
- Case $(\texttt{ABS}_{\mathcal{S}})$. $\dfrac{e \xrightarrow{n+1} e'}{\lambda x.e \xrightarrow{n+1} \lambda x.e'}$ By (IH), $depth(e) = n{+}1$. Then, $depth(\lambda x.e) = depth(e) = n + 1$.

□

The translation rule creates a new continuation for each `unbox` and stores it in the continuation stack K. Later, the rule consumes the last continuation of the stack for each `box`.

Lemma 2 states that if the translation of e results in the continuation stack K, the length $|K|$ of the continuation stack equals to the depth $depth(e)$ of e, which is the number of nested `unbox`es in e that is not enclosed by `box`es.

Definition 2 (Continuation Stack Length)**.** *$|K|$ denotes the length of the continuation stack K, defined as the following.*

$$\begin{array}{ll} |\bot| & = 0 \\ |K, \kappa| & = |K| + 1 \end{array}$$

Clearly, $|K| \geq 0$.

Lemma 2 (Continuation Stack Length wrt Depth)**.** *If $e \in Expr_{\mathcal{S}}$ and $R \vdash e \mapsto (\underline{e}, K)$ then $|K| = depth(e)$.*

Proof. By structural induction on the derivation of $\cdot \vdash \cdot \mapsto (\cdot, \cdot)$ (IH).

- Case $(\texttt{TCON})$. $R \vdash i \mapsto (i, \bot)$. $|K| = |\bot| = 0 = depth(i)$.
- Case $(\texttt{TVAR})$. $R \vdash x \mapsto (x, \bot)$. $|K| = |\bot| = 0 = depth(x)$.
- Case $(\texttt{TABS})$. $\dfrac{R, (r_n; x) \vdash e \mapsto (\underline{e}, K)}{R, r_n \vdash \lambda x.e \mapsto (\lambda x.\underline{e}, K)}$. By (IH), $|K| = depth(e)$. Then, $depth(\lambda x.e) = depth(e) = |K|$.
- Case $(\texttt{TAPP})$. $\dfrac{R \vdash e_1 \mapsto (\underline{e_1}, K_1) \quad R \vdash e_2 \mapsto (\underline{e_2}, K_2)}{R \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2)}$. By (IH), $|K_1| = depth(e_1)$. By (IH), $|K_2| = depth(e_2)$. Then, $depth(e_1\ e_2) = max(depth(e_1), depth(e_2)) = max(|K_1|, |K_2|) = |K_1 \bowtie K_2|$.
- Case $(\texttt{TBOX})_1$. $\dfrac{R, \bot \vdash e \mapsto (\underline{e}, (K, \kappa))}{R \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], K)}$. By (IH), $|K, \kappa| = |K| + 1 = depth(e)$. Then, $depth(\texttt{box}\ e) = depth(e) - 1 = |K| + 1 - 1 = |K|$.

- Case $(\texttt{TBOX})_2$. $\dfrac{R, \bot \vdash e \mapsto (\underline{e}, \bot)}{R \vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot)}$. By (IH), $|K| = |\bot| = 0 = \mathit{depth}(e)$. Then, $\mathit{depth}(\texttt{box}\ e) = 0$.

- Case $(\texttt{TUNB})$. $\dfrac{R, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, K) \qquad \nu = \{x/x \mid x \in r_n\} \qquad \texttt{new}\ H}{R, r_{n-1}, r_n \vdash \texttt{unbox}\ e \mapsto (H^\nu\texttt{()}, (K, (\delta H.[\cdot]) \odot_\nu \underline{e}))}$. By (IH), $|K| = \mathit{depth}(e)$. Then, $\mathit{depth}(\texttt{unbox}\ e) = \mathit{depth}(e) + 1 = |K| + 1 = |K, (\delta H.[\cdot]) \odot_\nu \underline{e}|$.

- Case $(\texttt{TRUN})$. $\dfrac{R \vdash e \mapsto (\underline{e}, K)}{R \vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()}), K)}$. By (IH), $|K| = \mathit{depth}(e)$. Then, $\mathit{depth}(\texttt{run}\ e) = \mathit{depth}(e) = |K|$.

□

Corollary 1. *Let $e \in \mathit{Expr}_\mathcal{S}$ and $R \vdash e \mapsto (\underline{e}, K)$ and $e \xrightarrow{n} e'$. Then $|K| = n$.*

Proof. Corollary of Lemma 1 and Lemma 2. □

Lemma 3 (Hole Variable Binding)**.** *The n-stage hole variables are bound at the stage $n-1$.*

- *Suppose $R \vdash e \mapsto (\underline{e}, (\kappa_1, \cdots, \kappa_{k-1}, \kappa_k, \cdots, \kappa_n))$.*
  - $\mathit{FH}(e) \subseteq \mathit{BH}(\kappa_n)$
  - $\mathit{FH}(\kappa_k) \subseteq \mathit{BH}(\kappa_{k-1}) \quad (2 \le k \le n)$
  - $\mathit{FH}(\kappa_1) = \varnothing$
- *Suppose $R \vdash e \mapsto (\underline{e}, (\kappa_1, \cdots, \kappa_{k-1}, \kappa_k[\kappa'_k], \cdots, \kappa_n))$.*
  - $\mathit{FH}(\kappa'_k) \cap \mathit{BH}(\kappa_k) = \varnothing$
  - $\mathit{FH}(\kappa'_k) \subseteq \mathit{BH}(\kappa_{k-1}) \quad (2 \le k \le n)$

Proof. By structural induction on $\mathit{Expr}_\mathcal{S}$. □

5.3 Existence of Translation

Lemma 4 states that for any expression e of $\lambda_\mathcal{S}$ whose depth $\mathit{depth}(e) = n$, there exists a translation image $\underline{e}$ of e, if a proper environment stack is provided. Here 'proper' means that the length of the environment stack is greater than n.

Furthermore, Lemma 5 states that the translation depends on only last n environments of the stack.

Lemma 4 (Existence)**.** *If $e \in \mathit{Expr}_\mathcal{S}$ and $\mathit{depth}(e) = n$ then for all $r_i \in \mathcal{L}(\mathit{Var})$ $(0 \le i \le n)$, there exists $\underline{e}$ and K such that $r_0, \cdots, r_n \vdash e \mapsto (\underline{e}, K)$.*

Proof. By structural induction on $\mathit{Expr}_\mathcal{S}$ (IH).

- Case $e = i$. $\mathit{depth}(i) = 0$. By $(\texttt{TCON})$, for all $r_0 \in \wp(\mathit{Var})$, we can have $r_0 \vdash i \mapsto (i, \bot)$.

- Case $e = x$. $\mathit{depth}(x) = 0$. By $(\texttt{TVAR})$, for all $r_0 \in \wp(\mathit{Var})$, we can have $r_0 \vdash x \mapsto (x, \bot)$.

- Case $e = \lambda x.e'$. By (IH), for all $r_0, \cdots, r_{n-1}, r_n \in \wp(\mathit{Var})$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_{n-1}, r_n \vdash e' \mapsto (\underline{e'}, K)$. Then, for all $r_0, \cdots, r_{n-1}, r'_n \in \wp(\mathit{Var})$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_{n-1}, (r'_n; x) \vdash e' \mapsto (\underline{e'}, K)$. Then, by $(\texttt{TABS})$, for all $r_0, \cdots, r_{n-1}, r'_n \in \wp(\mathit{Var})$, we can have $r_0, \cdots, r_{n-1}, r'_n \vdash \lambda x.e' \mapsto (\lambda x.\underline{e'}, K)$.

- Case $e = e_1\ e_2$. By (IH), for all $r_i \in \wp(\mathit{Var})$ $(0 \le i \le n)$, there exists $\underline{e_1}$ and K_1 such that $r_0, \cdots, r_n \vdash e_1 \mapsto (\underline{e_1}, K_1)$. By (IH), for all $r_i \in \wp(\mathit{Var})$ $(0 \le i \le n)$, there exists $\underline{e_2}$ and K_2 such that $r_0, \cdots, r_n \vdash e_2 \mapsto (\underline{e_2}, K_2)$. Then, by $(\texttt{TAPP})$, for all $r_i \in \wp(\mathit{Var})$ $(0 \le i \le n)$, we can have $r_0, \cdots, r_n \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2)$.

- Case $e = \texttt{box}\ e'$.
  - Case $depth(e') = 0$. $depth(\texttt{box}\ e') = 0$. By (IH), for all $r_0 \in \wp(Var)$, there exists $\underline{e'}$ and K such that $r_0 \vdash e' \mapsto (\underline{e'}, K)$. By Lemma 5, for all $r'_0, r_0 \in \wp(Var)$, $r'_0, r_0 \vdash e' \mapsto (\underline{e'}, K)$. Since $depth(e') = 0$, by Lemma 2, $|K| = 0$, that is, $K = \bot$. Therefore, for all $r'_0 \in \wp(Var)$, we can have $r'_0, \bot \vdash e' \mapsto (\underline{e'}, \bot)$. By $(\texttt{TBOX})_2$, for all $r'_0 \in \wp(Var)$, we can have $r'_0 \vdash \texttt{box}\ e' \mapsto (\lambda u.\underline{e'}, \bot)$.
  - Case $depth(e') \geq 1$. $n = depth(\texttt{box}\ e') = depth(e') - 1$. That is, $depth(e') = n + 1$. By (IH), for all $r_0, \cdots, r_n, r_{n+1} \in \wp(Var)$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_n, r_{n+1} \vdash e' \mapsto (\underline{e'}, K)$. Since $depth(e') = n + 1$, by Lemma 2, $|K| = n + 1 > 0$. Then, let $K = (K', \kappa')$. Then, for all $r_0, \cdots, r_n \in \wp(Var)$, there exists $\underline{e'}$, K' and κ' such that $r_0, \cdots, r_n, \bot \vdash e' \mapsto (\underline{e'}, (K', \kappa'))$. By $(\texttt{TBOX})_1$, for all $r_0, \cdots, r_n \in \wp(Var)$, we can have $r_0, \cdots, r_n, \bot \vdash \texttt{box}\ e' \mapsto (\kappa'[\lambda u.\underline{e'}], K')$.
- Case $e = \texttt{unbox}\ e'$. $n = depth(\texttt{unbox}\ e') = depth(e') + 1$. That is, $depth(e') = n - 1$. By (IH), for all $r_0, \cdots, r_{n-2}, r_{n-1} \in \wp(Var)$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_{n-2}, r_{n-1} \vdash e' \mapsto (\underline{e'}, K)$. Then, for all $r_0, \cdots, r_{n-2}, r'_{n-1}, r'_n \in \wp(Var)$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_{n-2}, (r'_{n-1}; r'_n) \vdash e' \mapsto (\underline{e'}, K)$. Then, by $(\texttt{TUNB})$, for all $r_0, \cdots, r_{n-2}, r'_{n-1}, r'_n \in \wp(Var)$, we can have $r_0, \cdots, r_{n-2}, r'_{n-1}, r'_n \vdash \texttt{unbox}\ e' \mapsto (H^\nu\texttt{()}, (K, (\delta H.[\cdot]) \odot_\nu \underline{e'}))$, where $\nu = \{x/x \mid x \in r'_n\}$.
- Case $e = \texttt{run}\ e'$. By (IH), for all $r_i \in \wp(Var)$ $(0 \leq i \leq n)$, there exists $\underline{e'}$ and K such that $r_0, \cdots, r_n \vdash e' \mapsto (\underline{e'}, K)$. Then, by $(\texttt{TRUN})$, for all $r_i \in \wp(Var)$ $(0 \leq i \leq n)$, we can have $r_0, \cdots, r_n \vdash \texttt{run}\ e' \mapsto (\texttt{let}\ h = \underline{e'}\ \texttt{in}\ (h\texttt{()}), K)$.

□

Lemma 5 (Weakening)**.** *Suppose $e \in Expr_{\mathcal{S}}$ and $depth(e) = k$. If $\overbrace{r'_1, \cdots, r'_n}, r_1, \cdots, r_k \vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k)$, where $n \geq 1$, then for all $r''_1, \cdots, r''_m \in \mathcal{L}(Var)$, where $m \geq 1$, we have $\underbrace{r''_1, \cdots, r''_m}, r_1, \cdots, r_k \vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k)$.*

Proof. By structural induction on the derivation of $\cdot \vdash \cdot \mapsto (\cdot, \cdot)$ (IH).

- Case $(\texttt{TCON})$. $k = depth(i) = 0$. By $(\texttt{TCON})$, $\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash i \mapsto (i, \bot)$.
- Case $(\texttt{TVAR})$. $k = depth(x) = 0$. By $(\texttt{TVAR})$, $\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash x \mapsto (x, \bot)$.
- Case $(\texttt{TABS})$.
  - Case $k = 0$. $k = depth(\lambda x.e) = depth(e) = 0$.

$$
\begin{aligned}
r'_1, \cdots, r'_n &\vdash \lambda x.e \mapsto (\lambda x.\underline{e}, \bot) && \text{(Premise)}\\
r'_1, \cdots, (r'_n; x) &\vdash e \mapsto (\underline{e}, \bot) && ((\texttt{TABS}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m &\vdash e \mapsto (\underline{e}, \bot) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, (r''_m; x) &\vdash e \mapsto (\underline{e}, \bot) &&\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m &\vdash \lambda x.e \mapsto (\lambda x.\underline{e}, \bot) && ((\texttt{TABS}))
\end{aligned}
$$

  - Case $k \geq 1$. $k = depth(\lambda x.e) = depth(e) \geq 1$.

$$
\begin{aligned}
r'_1, \cdots, r'_n, r_1, \cdots, r_k &\vdash \lambda x.e \mapsto (\lambda x.\underline{e}, \kappa_1, \cdots, \kappa_k) && \text{(Premise)}\\
r'_1, \cdots, r'_n, r_1, \cdots, (r_k; x) &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k) && ((\texttt{TABS}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, (r_k; x) &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k &\vdash \lambda x.e \mapsto (\lambda x.\underline{e}, \kappa_1, \cdots, \kappa_k) && ((\texttt{TABS}))
\end{aligned}
$$

- Case (TAPP).
  - Case $k = 0$. $k = depth(e_1\ e_2) = max(depth(e_1), depth(e_2)) = 0$. Then, $depth(e_1) = depth(e_2) = 0$.

$$r'_1, \cdots, r'_n \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, \bot) \quad \text{(Premise)}$$
$$r'_1, \cdots, r'_n \vdash e_1 \mapsto (\underline{e_1}, \bot) \quad ((\texttt{TAPP}))$$
$$r'_1, \cdots, r'_n \vdash e_2 \mapsto (\underline{e_2}, \bot) \quad ((\texttt{TAPP}))$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash e_1 \mapsto (\underline{e_1}, \bot) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash e_2 \mapsto (\underline{e_2}, \bot) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, \bot) \quad ((\texttt{TAPP}))$$

  - Case $k \geq 1$. $k = depth(e_1\ e_2) = max(depth(e_1), depth(e_2)) \geq 1$. $|K_1 \bowtie K_2| = max(|K_1|, |K_2|) = k$.

$$r'_1, \cdots, r'_n, r_1, \cdots, r_k \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2) \quad \text{(Premise)}$$
$$r'_1, \cdots, r'_n, r_1, \cdots, r_k \vdash e_1 \mapsto (\underline{e_1}, K_1) \quad ((\texttt{TAPP}))$$
$$r'_1, \cdots, r'_n, r_1, \cdots, r_k \vdash e_2 \mapsto (\underline{e_2}, K_2) \quad ((\texttt{TAPP}))$$
$$\forall r''_1, \cdots, r''_m, r'''_1, \cdots, r'''_{k-|K_1|} \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r'''_1, \cdots, r'''_{k-|K_1|}, r_{k-|K_1|+1}, \cdots, r_k \vdash e_1 \mapsto (\underline{e_1}, K_1) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m, r'''_1, \cdots, r'''_{k-|K_2|} \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r'''_1, \cdots, r'''_{k-|K_2|}, r_{k-|K_2|+1}, \cdots, r_k \vdash e_2 \mapsto (\underline{e_2}, K_2) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k \vdash e_1 \mapsto (\underline{e_1}, K_1)$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k \vdash e_2 \mapsto (\underline{e_2}, K_2)$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2) \quad ((\texttt{TAPP}))$$

- Case $(\texttt{TBOX})_1$.
  - Case $k = 0$. $k = depth(\texttt{box}\ e) = 0$.

$$r'_1, \cdots, r'_n \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], \bot) \quad \text{(Premise)}$$
$$r'_1, \cdots, r'_n, \bot \vdash e \mapsto (\underline{e}, \kappa) \quad ((\texttt{TBOX})_1)$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, \bot \vdash e \mapsto (\underline{e}, \kappa) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], \bot) \quad ((\texttt{TBOX})_1)$$

  - Case $k \geq 1$. $k = depth(\texttt{box}\ e) \geq 1$. Then, $depth(e) \geq 2$, since $depth(\texttt{box}\ e) = depth(e) - 1$.

$$r'_1, \cdots, r'_n, r_1, \cdots, r_k \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], \kappa_1, \cdots, \kappa_k) \quad \text{(Premise)}$$
$$r'_1, \cdots, r'_n, r_1, \cdots, r_k, \bot \vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k, \kappa) \quad ((\texttt{TBOX})_1)$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k, \bot \vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k, \kappa) \quad \text{(IH)}$$
$$\forall r''_1, \cdots, r''_m \in \mathcal{L}(Var).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], \kappa_1, \cdots, \kappa_k) \quad ((\texttt{TBOX})_1)$$

- Case $(\texttt{TBOX})_2$.

$$
\begin{aligned}
r'_1, \cdots, r'_n &\vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot) && \text{(Premise)}\\
r'_1, \cdots, r'_n, \bot &\vdash e \mapsto (\underline{e}, \bot) && ((\texttt{TBOX})_1)\\
\forall r''_1, \cdots, r''_m, r''' \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r''' &\vdash e \mapsto (\underline{e}, \bot) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, \bot &\vdash e \mapsto (\underline{e}, \bot) &&\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m &\vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot) && ((\texttt{TBOX})_1)
\end{aligned}
$$

- Case $(\texttt{TUNB})$. $k = \mathit{depth}(\texttt{unbox}\ e) \geq 1$.
  - Case $k = 1$. $k = \mathit{depth}(\texttt{unbox}\ e) = \mathit{depth}(e) + 1 = 1$. Then $\mathit{depth}(e) = 0$.

$$
\begin{aligned}
r'_1, \cdots, r'_n, r_1 &\vdash \texttt{unbox}\ e \mapsto (H^\nu(), (\delta H.[\cdot]) \odot_\nu \underline{e}) && \text{(Premise)}\\
r'_1, \cdots, (r'_n; r_1) &\vdash e \mapsto (\underline{e}, \bot) && ((\texttt{TUNB}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m &\vdash e \mapsto (\underline{e}, \bot) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, (r''_m; r_1) &\vdash e \mapsto (\underline{e}, \bot) &&\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r_1 &\vdash \texttt{unbox}\ e \mapsto (H^\nu(), (\delta H.[\cdot]) \odot_\nu \underline{e}) && \text{(Premise)}
\end{aligned}
$$

  - Case $k \geq 2$. $k = \mathit{depth}(\texttt{unbox}\ e) = \mathit{depth}(e) + 1 \geq 2$. Then, $\mathit{depth}(e) \geq 1$.

$$
\begin{aligned}
r'_1, \cdots, r'_n, r_1, \cdots, r_{k-1}, r_k &\vdash \texttt{unbox}\ e \mapsto (H^\nu(), \kappa_1, \cdots, \kappa_{k-1}, ((\delta H.[\cdot]) \odot_\nu \underline{e})) && \text{(Premise)}\\
r'_1, \cdots, r'_n, r_1, \cdots, (r_{k-1}; r_k) &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_{k-1}) && ((\texttt{TUNB}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r_1, \cdots, (r_{k-1}; r_k) &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_{k-1}) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r_1, \cdots, r_{k-1}, r_k &\vdash \texttt{unbox}\ e \mapsto (H^\nu(), \kappa_1, \cdots, \kappa_{k-1}, ((\delta H.[\cdot]) \odot_\nu \underline{e})) && ((\texttt{TUNB}))
\end{aligned}
$$

- Case $(\texttt{TRUN})$.
  - Case $k = 0$. $k = \mathit{depth}(\texttt{unbox}\ e) = \mathit{depth}(e) = 0$.

$$
\begin{aligned}
r'_1, \cdots, r'_n &\vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h()), \bot) && \text{(Premise)}\\
r'_1, \cdots, r'_n &\vdash e \mapsto (\underline{e}, \bot) && ((\texttt{TRUN}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m &\vdash e \mapsto (\underline{e}, \bot) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m &\vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h()), \bot) && ((\texttt{TRUN}))
\end{aligned}
$$

  - Case $k \geq 1$. $k = \mathit{depth}(\texttt{unbox}\ e) = \mathit{depth}(e) \geq 1$.

$$
\begin{aligned}
r'_1, \cdots, r'_n, r_1, \cdots, r_k &\vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h()), \kappa_1, \cdots, \kappa_k) && \text{(Premise)}\\
r'_1, \cdots, r'_n, r_1, \cdots, r_k &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k) && ((\texttt{TRUN}))\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k &\vdash e \mapsto (\underline{e}, \kappa_1, \cdots, \kappa_k) && \text{(IH)}\\
\forall r''_1, \cdots, r''_m \in \mathcal{L}(\mathit{Var}).\ r''_1, \cdots, r''_m, r_1, \cdots, r_k &\vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h()), \kappa_1, \cdots, \kappa_k) && ((\texttt{TRUN}))
\end{aligned}
$$

□

5.4 Value Preservation

Lemma 6 states that a value v of $\lambda_{\mathcal{S}}$ translates to a value of $\lambda_{\mathcal{C}}$.

Lemma 6 (Value Preservation)**.** *If* $v \in Value^0_{\mathcal{S}}$ *and* $R \vdash v \mapsto (\underline{v}, K)$ *then* $K = \bot$ *and* $\underline{v} \in Value_{\mathcal{C}}$*.*

Proof. By case analysis on $Value^0_{\mathcal{S}}$.

- Case $v = i$. By (TCON), $R \vdash i \mapsto (i, \bot)$. $K = \bot$. $i \in Value_{\mathcal{C}}$.
- Case $v = x$. By (TVAR), $R \vdash x \mapsto (x, \bot)$. $K = \bot$. $x \in Value_{\mathcal{C}}$.
- Case $v = \lambda x.e^0$. By (TABS), $\dfrac{R, (r_n; x) \vdash e^0 \mapsto (\underline{e^0}, K)}{R, r_n \vdash \lambda x.e^0 \mapsto (\lambda x.\underline{e^0}, K)}$. By Lemma 1, $depth(e^0) \le 0$, since $e^0 \in Expr^0_{\mathcal{S}}$. Then, $depth(e^0) = 0$. By Lemma 2, $|K| = depth(e^0) = 0$. Then, $K = \bot$. $\lambda x.\underline{e^0} \in Value_{\mathcal{C}}$.
- Case $v = \texttt{box}\ v^1$. There are two rules: $(\texttt{TBOX})_1$ and $(\texttt{TBOX})_2$. However, By Lemma 1 and Lemma 2, if $R \vdash v^1 \mapsto (\underline{v^1}, K)$, then $K = \bot$. Therefore, $(\texttt{TBOX})_1$ cannot be applied, since $(K, \kappa) \ne \bot$. By $(\texttt{TBOX})_2$, $\dfrac{R, \bot \vdash v^1 \mapsto (\underline{v^1}, \bot)}{R \vdash \texttt{box}\ v^1 \mapsto (\lambda u.\underline{v^1}, \bot)}$. $K = \bot$. $\lambda u.\underline{v^1} \in Value_{\mathcal{C}}$.

□

5.5 Substitution Preservation

In Lemma 7, $\{v/x\}\kappa_i = \kappa_i$ means that every occurence of x is bound in every κ_i, i.e., $x \notin FV(\kappa_k, \cdots, \kappa_n)$. Note that it does not mean x does not occur in κ_i.

Lemma 7 (Free Variables Preservation)**.** *Suppose* $v \in Value_{\mathcal{C}}$, $x \in Var$, $n \ge 1$, $m \ge 0$ *and* $1 \le k \le n$*. If*

$$x \in r_n \qquad \textit{and} \qquad r_0, \cdots, \underbrace{r_n}, r_{n+1}, \cdots, r_{n+m} \vdash e \mapsto (\underline{e}, (\underbrace{\kappa_k, \cdots, \kappa_n}, \kappa_{n+1}, \cdots, \kappa_{n+m}))$$

then

$$\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$$

Proof. By structural induction on the derivation of $\cdot \vdash \cdot \mapsto (\cdot, \cdot)$ (IH).

- Case (TCON). $K = \bot$. Premise is not satisfied. Not applicable case. Ignored.
- Case (TVAR). $K = \bot$. Premise is not satisfied. Not applicable case. Ignored.
- Case (TABS).
  - $m = 0$. $x \in r_n$ and

    $$\frac{r_0, \cdots, (r_n; y) \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n))}{r_0, \cdots, r_n \vdash \lambda y.e \mapsto (\lambda y.\underline{e}, (\kappa_k, \cdots, \kappa_n))}$$

    Clearly, $x \in (r_n; y)$. By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.
  - $m \ge 1$. $x \in r_n$ and

    $$\frac{r_0, \cdots, r_n, r_{n+1}, \cdots, (r_{n+m}; y) \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash \lambda y.e \mapsto (\lambda y.\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}$$

    By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.

- Case (TAPP). Since $|K| = |K_1 \bowtie K_2| = max(|K_1|, |K_2|)$, $|K_1| = |K|$ or $|K_2| = |K|$.
  - Case $|K_1| = |K|$. $|K_2| \leq |K| = n + m - k + 1$.
    - Case $m = 0$.
      - Case $|K_2| > 0$. Let

$$\begin{aligned} K &= \kappa_k, \cdots, \kappa_n \\ K_1 &= \kappa'_k, \cdots, \kappa'_n \\ K_2 &= \kappa''_{k+l}, \cdots, \kappa''_n \qquad (0 \leq l \leq n-k) \end{aligned}$$

By (TAPP),

$$\frac{r_0, \cdots, r_n \vdash e_1 \mapsto (\underline{e_1}, (\kappa'_k, \cdots, \kappa'_n)) \qquad r_0, \cdots, r_n \vdash e_2 \mapsto (\underline{e_2}, (\kappa''_{k+l}, \cdots, \kappa''_n))}{r_0, \cdots, r_n \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, (\kappa_k, \cdots, \kappa_n))}$$

By (IH), $\{v/x\}\kappa'_i = \kappa'_i$ $(k \leq i \leq n)$ and $\{v/x\}\kappa''_i = \kappa''_i$ $(k + l \leq i \leq n)$. Furthermore,

$$\begin{aligned} K &= K_1 \bowtie K_2 \\ \kappa_k, \cdots, \kappa_n &= (\kappa'_k, \cdots, \kappa'_n) \bowtie (\kappa''_{k+l}, \cdots, \kappa''_n) \qquad (0 \leq l \leq n-k) \\ &= \kappa'_k, \cdots, \kappa'_{k+l-1}, \kappa'_{k+l}[\kappa''_{k+l}], \cdots, \kappa'_n[\kappa''_n] \end{aligned}$$

That is,

$$\kappa_i = \begin{cases} \kappa'_i & \text{if } k \leq i \leq k+l-1 \\ \kappa'_i[\kappa''_i] & \text{if } k+l \leq i \leq n \end{cases}$$

Therefore,

$$\begin{aligned} \{v/x\}\kappa_i &= \begin{cases} \{v/x\}\kappa'_i & \text{if } k \leq i \leq k+l-1 \\ \{v/x\}\kappa'_i[\{v/x\}\kappa''_i] & \text{if } k+l \leq i \leq n \end{cases} & (k \leq i \leq n) \\ &= \begin{cases} \kappa'_i & \text{if } k \leq i \leq k+l-1 \\ \kappa'_i[\kappa''_i] & \text{if } k+l \leq i \leq n \end{cases} & (k \leq i \leq n) \quad \text{(IH)} \\ &= \kappa_i & (k \leq i \leq n) \end{aligned}$$

      - Case $|K_2| = 0$. Let

$$\begin{aligned} K &= \kappa_k, \cdots, \kappa_n \\ K_1 &= \kappa'_k, \cdots, \kappa'_n \\ K_2 &= \bot \end{aligned}$$

By (TAPP),

$$\frac{r_0, \cdots, r_n \vdash e_1 \mapsto (\underline{e_1}, (\kappa'_k, \cdots, \kappa'_n)) \qquad r_0, \cdots, r_n \vdash e_2 \mapsto (\underline{e_2}, \bot)}{r_0, \cdots, r_n \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, (\kappa_k, \cdots, \kappa_n))}$$

By (IH), $\{v/x\}\kappa'_i = \kappa'_i$ $(k \leq i \leq n)$. Furthermore,

$$\begin{aligned} K &= K_1 \bowtie K_2 \\ \kappa_k, \cdots, \kappa_n &= (\kappa'_k, \cdots, \kappa'_n) \bowtie \bot \\ &= \kappa'_k, \cdots, \kappa'_n \end{aligned}$$

That is,

$$\kappa_i = \kappa'_i \quad (k \leq i \leq n)$$

Therefore,

$$\begin{aligned}\{v/x\}\kappa_i &= \{v/x\}\kappa_i' && (k \le i \le n)\\ &= \kappa_i' && (k \le i \le n) && \text{(IH)}\\ &= \kappa_i && (k \le i \le n)\end{aligned}$$

* Case $m \ge 1$.
  - Case $|K_2| > m$. Let

$$\begin{aligned}K &= \kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}\\ K_1 &= \kappa_k', \cdots, \kappa_n', \kappa_{n+1}', \cdots, \kappa_{n+m}'\\ K_2 &= \kappa_{k+l}'', \cdots, \kappa_n'', \kappa_{n+1}'', \cdots, \kappa_{n+m}'' \qquad (0 \le l \le n-k)\end{aligned}$$

By (TAPP),

$$\frac{\begin{array}{c} r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_1 \mapsto (\underline{e_1}, (\kappa_k', \cdots, \kappa_n', \kappa_{n+1}', \cdots, \kappa_{n+m}'))\\ r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_2 \mapsto (\underline{e_2}, (\kappa_{k+l}'', \cdots, \kappa_n'', \kappa_{n+1}'', \cdots, \kappa_{n+m}''))\end{array}}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}$$

By (IH), $\{v/x\}\kappa_i' = \kappa_i'$ $(k \le i \le n)$ and $\{v/x\}\kappa_i'' = \kappa_i''$ $(k+l \le i \le n)$. Furthermore,

$$\begin{aligned}K &= K_1 \bowtie K_2\\ \kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m} &= (\kappa_k', \cdots, \kappa_n', \kappa_{n+1}', \cdots, \kappa_{n+m}')\\ &\quad \bowtie (\kappa_{k+l}'', \cdots, \kappa_n'', \kappa_{n+1}'', \cdots, \kappa_{n+m}'') \qquad (0 \le l \le n-k)\\ &= \kappa_k', \cdots, \kappa_{k+l-1}', \kappa_{k+l}'[\kappa_{k+l}''], \cdots, \kappa_n'[\kappa_n''], \kappa_{n+1}'[\kappa_{n+1}''], \cdots, \kappa_{n+m}'[\kappa_{n+m}'']\end{aligned}$$

That is,

$$\kappa_i = \begin{cases} \kappa_i' & \text{if } k \le i \le k+l-1\\ \kappa_i'[\kappa_i''] & \text{if } k+l \le i \le n+m\end{cases}$$

Therefore,

$$\begin{aligned}\{v/x\}\kappa_i &= \begin{cases} \{v/x\}\kappa_i' & \text{if } k \le i \le k+l-1\\ \{v/x\}\kappa_i'[\{v/x\}\kappa_i''] & \text{if } k+l \le i \le n\end{cases} && (k \le i \le n)\\ &= \begin{cases} \kappa_i' & \text{if } k \le i \le k+l-1\\ \kappa_i'[\kappa_i''] & \text{if } k+l \le i \le n\end{cases} && (k \le i \le n) \quad \text{(IH)}\\ &= \kappa_i && (k \le i \le n)\end{aligned}$$

  - Case $|K_2| \le m$. Let

$$\begin{aligned}K &= \kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}\\ K_1 &= \kappa_k', \cdots, \kappa_n', \kappa_{n+1}', \cdots, \kappa_{n+m}'\\ K_2 &= \kappa_{n+l}'', \cdots, \kappa_{n+m}'' \qquad (1 \le l \le m)\end{aligned}$$

By (TAPP),

$$\frac{\begin{array}{c} r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_1 \mapsto (\underline{e_1}, (\kappa_k', \cdots, \kappa_n', \kappa_{n+1}', \cdots, \kappa_{n+m}'))\\ r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_2 \mapsto (\underline{e_2}, (\kappa_{n+l}'', \cdots, \kappa_{n+m}''))\end{array}}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}$$

By (IH), $\{v/x\}\kappa'_i = \kappa'_i \ \ (k \le i \le n)$. Furthermore,

$$\begin{aligned} K &= K_1 \bowtie K_2 \\ \kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m} &= \kappa'_k, \cdots, \kappa'_n, \kappa'_{n+1}, \cdots, \kappa'_{n+m} \bowtie \kappa''_{n+l}, \cdots, \kappa''_{n+m} \qquad (1 \le l \le m) \\ &= \kappa'_k, \cdots, \kappa'_n, \kappa'_{n+1}, \cdots, \kappa'_{n+l-1}, \kappa'_{n+l}[\kappa''_{n+l}], \cdots, \kappa'_{n+m}[\kappa''_{n+m}] \end{aligned}$$

That is,

$$\kappa_i = \begin{cases} \kappa'_i & \text{if } k \le i \le n+l-1 \\ \kappa'_i[\kappa''_i] & \text{if } n+l \le i \le n+m \end{cases}$$

Therefore,

$$\begin{aligned} \{v/x\}\kappa_i &= \{v/x\}\kappa'_i && (k \le i \le n) \\ &= \kappa'_i && (k \le i \le n) \qquad \text{(IH)} \\ &= \kappa_i && (k \le i \le n) \end{aligned}$$

 - Case $|K_2| = |K_1 \bowtie K_2|$. Dual with the previous case. Omitted.

- Case $(\texttt{TBOX})_1$.
  - Case $m = 0$. $x \in r_n$ and

$$\frac{r_0, \cdots, r_n, \bot \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa))}{r_0, \cdots, r_n \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], (\kappa_k, \cdots, \kappa_n))}$$

  By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.
  - Case $m \ge 1$. $x \in r_n$ and

$$\frac{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m}, \bot \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}, \kappa))}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}$$

  By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.

- Case $(\texttt{TBOX})_2$. $K = \bot$. Premise is not satisfied. Not applicable case. Ignored.
- Case $(\texttt{TUNB})$.
  - Case $m = 0$. $x \in r_n$ and

$$\frac{r_0, \cdots, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_{n-1})) \qquad r_n = x_l; \cdots; x_0 \qquad \nu = \{w_l/x_l, \cdots, w_0/x_0\}}{r_0, \cdots, r_{n-1}, r_n \vdash \texttt{unbox}\ e \mapsto (H^{\nu^{-1}}(), (\kappa_k, \cdots, \kappa_{n-1}, \underbrace{(\delta H.[\cdot]) \odot_\nu \underline{e}}_{\kappa_n}))}$$

  Clearly, $x \in (r_{n-1}; r_n)$. By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n-1)$. And

$$\begin{aligned} \{v/x\}\kappa_n &= \{v/x\}((\delta H.[\cdot]) \odot_\nu \underline{e}) \\ &= (\delta H.\{v/x\}[\cdot]) \odot_\nu \underline{e} \qquad (x \in dom(\nu)) \\ &= (\delta H.[\cdot]) \odot_\nu \underline{e} \\ &= \kappa_n \end{aligned}$$

  Therefore, $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.
  - Case $m = 1$. $x \in r_n$ and

$$\frac{r_0, \cdots, (r_n; r_{n+1}) \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n))}{r_0, \cdots, r_n, r_{n+1} \vdash \texttt{unbox}\ e \mapsto (H^{\nu^{-1}}(), (\kappa_k, \cdots, \kappa_n, (\delta H.[\cdot]) \odot_\nu \underline{e}))}$$

  Clearly, $x \in (r_n; r_{n+1})$. By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \le i \le n)$.

- Case $m \geq 2$. $x \in r_n$ and

$$\frac{r_0, \cdots, r_n, r_{n+1}, \cdots, (r_{n+m-1}; r_{n+m}) \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m-1}))}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m-1}, r_{n+m} \vdash \texttt{unbox}\ e \mapsto (H^{\nu^{-1}}\texttt{()}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m-1}, (\delta H.[\cdot]) \odot_\nu \underline{e}))}$$

By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \leq i \leq n)$.

- Case (TRUN).
  - Case $m = 0$. $x \in r_n$ and

$$\frac{r_0, \cdots, r_n \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n))}{r_0, \cdots, r_n \vdash \texttt{unbox}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()}), (\kappa_k, \cdots, \kappa_n))}$$

By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \leq i \leq n)$.

  - Case $m \geq 1$. $x \in r_n$ and

$$\frac{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash e \mapsto (\underline{e}, (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}{r_0, \cdots, r_n, r_{n+1}, \cdots, r_{n+m} \vdash \texttt{unbox}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()}), (\kappa_k, \cdots, \kappa_n, \kappa_{n+1}, \cdots, \kappa_{n+m}))}$$

By (IH), $\{v/x\}\kappa_i = \kappa_i \ \ (k \leq i \leq n)$.

□

Lemma 8 (Substitution Preservation)**.** *Suppose $x \in Var$, $v \in Value^0_{\mathcal{S}}$, $n \geq 0$, $x \notin r_i \ \ (0 \leq i \leq n)$, and $\bot \vdash v \mapsto (\underline{v}, \bot)$. Let $R = r_0, \cdots, r_n$. If $R \vdash e \mapsto (\underline{e}, K)$ then $R \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}K)$.*

Proof. By structural induction on the derivation of $\cdot \vdash \cdot \mapsto (\cdot, \cdot)$ (IH).

- Case (TCON).
  1. $R \vdash i \mapsto (i, \bot)$ — Premise
  2. $[x \mapsto v]i = i$
  3. $\{\underline{v}/x\}i = i$
  4. $\{\underline{v}/x\}\bot = \bot$
  5. $R \vdash [x \mapsto v]i \mapsto (\{\underline{v}/x\}\underline{i}, \{\underline{v}/x\}\bot)$ — (1), (2), (3), (4)
- Case (TVAR). $R \vdash y \mapsto (y, \bot)$.
  - Case $x = y$.
    1. $\bot \vdash v \mapsto (\underline{v}, \bot)$ — Premise
    2. $R \vdash v \mapsto (\underline{v}, \bot)$ — Lemma 5
    3. $[x \mapsto v]y = v$
    4. $\{\underline{v}/x\}y = \underline{v}$
    5. $\{\underline{v}/x\}\bot = \bot$
    6. $R \vdash [x \mapsto v]y \mapsto (\{\underline{v}/x\}\underline{y}, \{\underline{v}/x\}\bot)$ — (2), (3), (4), (5)
  - Case $x \neq y$.
    1. $R \vdash y \mapsto (y, \bot)$ — Premise
    2. $[x \mapsto v]y = y$
    3. $\{\underline{v}/x\}y = y$
    4. $\{\underline{v}/x\}\bot = \bot$
    5. $R \vdash [x \mapsto v]y \mapsto (\{\underline{v}/x\}\underline{y}, \{\underline{v}/x\}\bot)$ — (1), (2), (3), (4)

- Case (TABS). $R \vdash \lambda y.e \mapsto (\lambda y.\underline{e}, K)$.
  - Case $x = y$.
    1. $R, r_n \vdash \lambda x.e \mapsto (\lambda x.\underline{e}, K)$ — Premise
    2. $R, (r_n; x) \vdash e \mapsto (\underline{e}, K)$ — (TABS)
    3. $\{\underline{v}/x\}K = K$ — (2), Lemma 7
    4. $[x \mapsto v]\lambda x.e = \lambda x.e$
    5. $\{\underline{v}/x\}\lambda x.\underline{e} = \lambda x.\underline{e}$
    6. $R, r_n \vdash [x \mapsto v]\lambda x.e \mapsto (\{\underline{v}/x\}\lambda x.\underline{e}, \{\underline{v}/x\}K)$ — (1), (3), (4), (5)
  - Case $x \neq y$.
    1. $R, r_n \vdash \lambda y.e \mapsto (\lambda y.\underline{e}, K)$ — Premise
    2. $R, (r_n; y) \vdash e \mapsto (\underline{e}, K)$ — (TABS)
    3. $R, (r_n; y) \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}K)$ — (IH)
    4. $R, r_n \vdash \lambda y.[x \mapsto v]e \mapsto (\lambda y.\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}K)$ — (TABS)
    5. $R, r_n \vdash [x \mapsto v]\lambda y.e \mapsto (\{\underline{v}/x\}\lambda y.\underline{e}, \{\underline{v}/x\}K)$

- Case (TAPP).
  1. $R \vdash e_1\ e_2 \mapsto (\underline{e_1}\ \underline{e_2}, K_1 \bowtie K_2)$ — Premise
  2. $R \vdash e_1 \mapsto (\underline{e_1}, K_1)$ — (1), (TAPP)
  3. $R \vdash e_2 \mapsto (\underline{e_2}, K_2)$ — (1), (TAPP)
  4. $R \vdash [x \mapsto v]e_1 \mapsto (\{\underline{v}/x\}\underline{e_1}, \{\underline{v}/x\}K_1)$ — (2), (IH)
  5. $R \vdash [x \mapsto v]e_2 \mapsto (\{\underline{v}/x\}\underline{e_2}, \{\underline{v}/x\}K_2)$ — (3), (IH)
  6. $R \vdash [x \mapsto v]e_1\ [x \mapsto v]e_2 \mapsto (\{\underline{v}/x\}\underline{e_1}\ \{\underline{v}/x\}\underline{e_2}, \{\underline{v}/x\}K_1 \bowtie \{\underline{v}/x\}K_2)$ — (4), (5), (TAPP)
  7. $R \vdash [x \mapsto v](e_1\ e_2) \mapsto (\{\underline{v}/x\}(\underline{e_1}\ \underline{e_2}), \{\underline{v}/x\}(K_1 \bowtie K_2))$

- Case $(\texttt{TBOX})_1$.
  1. $R \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], K)$ — Premise
  2. $R, \bot \vdash e \mapsto (\underline{e}, (K, \kappa))$ — $(\texttt{TBOX})_1$
  3. $R, \bot \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}(K, \kappa))$ — (IH)
  4. $R, \bot \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, (\{\underline{v}/x\}K, \{\underline{v}/x\}\kappa))$
  5. $R \vdash \texttt{box}\ [x \mapsto v]e \mapsto ((\{\underline{v}/x\}\kappa)[\lambda u.\{\underline{v}/x\}\underline{e}], \{\underline{v}/x\}K)$ — $(\texttt{TBOX})_1$
  6. $R \vdash [x \mapsto v](\texttt{box}\ e) \mapsto (\{\underline{v}/x\}(\kappa[\lambda u.\underline{e}]), \{\underline{v}/x\}K)$

- Case $(\texttt{TBOX})_2$.
  1. $R \vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot)$ — Premise
  2. $R, \bot \vdash e \mapsto (\underline{e}, \bot)$ — $(\texttt{TBOX})_2$
  3. $R, \bot \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}\bot)$ — (IH)
  4. $R \vdash \texttt{box}\ [x \mapsto v]e \mapsto (\lambda u.\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}\bot)$ — $(\texttt{TBOX})_2$
  5. $R \vdash [x \mapsto v](\texttt{box}\ e) \mapsto (\{\underline{v}/x\}(\lambda u.\underline{e}), \{\underline{v}/x\}\bot)$

- Case (TUNB).
  1. $R, r_{n-1}, r_n \vdash \texttt{unbox}\ e \mapsto (H^\nu(), (K, (\delta H.[\cdot]) \odot_\nu \underline{e}))$ — Premise
  2. $R, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, K)$ — (TUNB)

3. $R, (r_{n-1}; r_n) \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}K)$ (IH)
4. $R, r_{n-1}, r_n \vdash \texttt{unbox}\ [x \mapsto v]e \mapsto (H^\nu(), (\{\underline{v}/x\}K, (\delta H.[\cdot]) \odot_\nu \{\underline{v}/x\}\underline{e}))$ (TUNB)
5. $R, r_{n-1}, r_n \vdash [x \mapsto v](\texttt{unbox}\ e) \mapsto (H^\nu(), (\{\underline{v}/x\}K, \{\underline{v}/x\}((\delta H.[\cdot]) \odot_\nu \underline{e})))$ $x \notin dom(\nu)$
6. $R, r_{n-1}, r_n \vdash [x \mapsto v](\texttt{unbox}\ e) \mapsto (H^\nu(), \{\underline{v}/x\}(K, (\delta H.[\cdot]) \odot_\nu \underline{e}))$

- Case (TRUN).
  1. $R \vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h()), K)$ Premise
  2. $R \vdash e \mapsto (\underline{e}, K)$ (TRUN)
  3. $R \vdash [x \mapsto v]e \mapsto (\{\underline{v}/x\}\underline{e}, \{\underline{v}/x\}K)$ (IH)
  4. $R \vdash \texttt{run}\ [x \mapsto v]e \mapsto (\texttt{let}\ h = \{\underline{v}/x\}\underline{e}\ \texttt{in}\ (h()), \{\underline{v}/x\}K)$ (TRUN)
  5. $R \vdash [x \mapsto v](\texttt{run}\ e) \mapsto (\{\underline{v}/x\}(\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h())), \{\underline{v}/x\}K)$

□

5.6 Simulation

Definition 3 (Admin-Normal Form)**.** *An expresion e is admin-normal form if and only if there does not exist e' such that $e \xrightarrow{\mathcal{A}} e'$. Furthermore, a continuation stack K is admin-normal form if and only if there does not exist K' such that $K \xrightarrow{\mathcal{A}} K'$.*

Lemma 9 (Admin-Normal)**.** *If $R \vdash e \mapsto (\underline{e}, K)$ then e and K are admin-normal form.*

Proof. By structural induction on $Expr_{\mathcal{S}}$ (IH). □

Definition 4 (Continuation Closure)**.** *Let $e \in Expr_{\mathcal{C}}$ and K be a continuation stack. The continuation closure $K(e)$ is defined as follows:*

$$K(e) = \begin{cases} K'(\kappa[e]) & \text{if } K = (K', \kappa) \\ e & \text{if } K = \bot \end{cases}$$

Lemma 10 (Free Hole Variables wrt Continuation Closure)**.** *If $e \in Expr_{\mathcal{S}}$ and $R \vdash e \mapsto (\underline{e}, K)$ then $FH(K(\underline{e})) = \varnothing$.*

Proof. By structural induction on $Expr_{\mathcal{S}}$ (IH). □

Remark 2. If $R \vdash e \mapsto (\underline{e}, K)$ then $\{()/u\}\underline{e} = \underline{e}$, where u is a variable newly introduced at $(\texttt{TBOX})_1$ and $(\texttt{TBOX})_2$.

Now we extend the reduction of $\lambda_{\mathcal{C}}$ into the translation as Figure 11.

Lemma 11 (Reduction Preservation)**.** *If $e \xrightarrow{n} e'$, $R \vdash e \mapsto (\underline{e}, K)$ and $R \vdash e' \mapsto (\underline{e'}, K')$, then $\langle K, \underline{e}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \underline{e'}\rangle$*

Proof. By structural induction on the derivation of $\cdot \xrightarrow{n} \cdot$ (IH).

- Case $(\texttt{APP}_{\mathcal{S}})_1$.
  - Case $n = 0$. $\dfrac{e_1 \xrightarrow{0} e_1'}{e_1\ e_2^0 \xrightarrow{0} e_1'\ e_2^0}$.

$$\frac{R \vdash e_1 \mapsto (\underline{e_1}, \bot) \quad R \vdash e_2^0 \mapsto (\underline{e_2^0}, \bot)}{R \vdash e_1\ e_2^0 \mapsto (\underline{e_1}\ \underline{e_2^0}, \bot)} \qquad \frac{R \vdash e_1' \mapsto (\underline{e_1'}, \bot) \quad R \vdash e_2^0 \mapsto (\underline{e_2^0}, \bot)}{R \vdash e_1'\ e_2^0 \mapsto (\underline{e_1'}\ \underline{e_2^0}, \bot)}$$

Definitions

$$
\begin{array}{rcl}
e & \in & \mathit{Expr}_{\mathcal{C}} \\
K & \in & \mathit{Continuation\ Stack} \\
\mathit{Modular\ Continuation\ Closure} & ::= & \langle K, e \rangle
\end{array}
$$

Reduction on Modular Continuation Closure

(RNOR)
$$\frac{e \xrightarrow{\mathcal{C};\mathcal{A}^*} e'}{\langle \bot, e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, e' \rangle}$$

$(\mathtt{RUNB})_1$
$$\frac{e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H \qquad n \geq 1}{\langle ((\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n), e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle ((\delta H.[\cdot]) \odot_\nu e'_H, \kappa_2, \cdots, \kappa_n), e \rangle}$$

$(\mathtt{RUNB})_2$
$$\frac{e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H \qquad n \geq 1}{\langle ((\delta H.\kappa) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n), e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle ((\delta H.\kappa) \odot_\nu e'_H, \kappa_2, \cdots, \kappa_n), e \rangle}$$

$(\mathtt{RHFE})_1$
$$\frac{(e[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} e' \qquad e_H \in \mathit{Value}_{\mathcal{C}}}{\langle (\delta H.[\cdot]) \odot_\nu e_H, e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, e' \rangle}$$

$(\mathtt{RHFE})_2$
$$\frac{(e[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} e' \qquad e_H \in \mathit{Value}_{\mathcal{C}}}{\langle (\delta H.\kappa) \odot_\nu e_H, e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \kappa, e' \rangle}$$

$(\mathtt{RHFK})_1$
$$\frac{(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa'_2 \qquad e_H \in \mathit{Value}_{\mathcal{C}} \qquad n \geq 2}{\langle ((\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n), e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (\kappa'_2, \kappa_3, \cdots, \kappa_n), e \rangle}$$

$(\mathtt{RHFK})_2$
$$\frac{(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa'_2 \qquad e_H \in \mathit{Value}_{\mathcal{C}} \qquad n \geq 2}{\langle ((\delta H.\kappa) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n), e \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (\kappa, \kappa'_2, \kappa_3, \cdots, \kappa_n), e \rangle}$$

Figure 11: Reduction on Modular Continuation Closure

1. $\langle \bot, \underline{e_1} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \underline{e_1'} \rangle$ (IH)
2. $\underline{e_1} \xrightarrow{\mathcal{C};\mathcal{A}^*} \underline{e_1'}$ (1), $(\texttt{RNOR})$
3. $\underline{e_1}\ \underline{e_2^0} \xrightarrow{\mathcal{C};\mathcal{A}^*} \underline{e_1'}\ \underline{e_2^0}$ (2), $(\texttt{APP}_{\mathcal{C}})_1$, $(\texttt{APP}_{\mathcal{A}})_1$
4. $\langle \bot, \underline{e_1}\ \underline{e_2^0} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \underline{e_1'}\ \underline{e_2^0} \rangle$ (3), $(\texttt{RNOR})$

– Case $n \geq 1$. $\dfrac{e_1 \xrightarrow{n} e_1'}{e_1\ e_2^n \xrightarrow{n} e_1'\ e_2^n}$.

$$\frac{R \vdash e_1 \mapsto (\underline{e_1}, K_1) \quad R \vdash e_2^n \mapsto (\underline{e_2^n}, K_2)}{R \vdash e_1\ e_2^n \mapsto (\underline{e_1}\ \underline{e_2^n}, K_1 \bowtie K_2)} \qquad \frac{R \vdash e_1' \mapsto (\underline{e_1'}, K_1') \quad R \vdash e_2^n \mapsto (\underline{e_2^n}, K_2)}{R \vdash e_1'\ e_2^n \mapsto (\underline{e_1'}\ \underline{e_2^n}, K_1' \bowtie K_2)}$$

By Lemma 1 and Lemma 2, $|K_1| = n$ and $|K_2| \leq n$. By (IH), $\langle K_1, \underline{e_1} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1', \underline{e_1'} \rangle$. In this case, $(\texttt{RNOR})$cannot be applied.

∗ Case $(\texttt{RUNB})_1$.
1. $K_1 = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RUNB})_1$
2. $K_1' = (\delta H.[\cdot]) \odot_\nu e_H', \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RUNB})_1$
3. $\underline{e_1} = \underline{e_1'}$ (IH), $(\texttt{RUNB})_1$
4. $\underline{e_1}\ \underline{e_2^n} = \underline{e_1'}\ \underline{e_2^n}$ (3)
5. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e_H'$ (IH), $(\texttt{RUNB})_1$
6. If $|K_2| = n$ then $\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n} \rangle$ (1), (2), (4), (5), $(\texttt{RUNB})_2$
7. If $|K_2| < n$ then $\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n} \rangle$ (1), (2), (4), (5), $(\texttt{RUNB})_1$

∗ Case $(\texttt{RUNB})_2$. Similar with $(\texttt{RUNB})_1$. Omitted.

∗ Case $(\texttt{RHFE})_1$.
1. $K_1 = (\delta H.[\cdot]) \odot_\nu e_H$ (IH), $(\texttt{RHFE})_1$
2. $K_1' = \bot$ (IH), $(\texttt{RHFE})_1$
3. $(\underline{e_1}[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e_1'}$ (IH), $(\texttt{RHFE})_1$
4. $((\underline{e_1}\ \underline{e_2^n})[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e_1'}\ \underline{e_2^n}$ (3), $H \notin FH(\underline{e_2^n})$
5. If $|K_2| = |K_1|$ then $\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n} \rangle$ (1), (2), (4), $(\texttt{RHFE})_2$
6. If $|K_2| < |K_1|$ then $\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n} \rangle$ (1), (2), (4), $(\texttt{RHFE})_1$

∗ Case $(\texttt{RHFE})_2$. Similar with $(\texttt{RHFE})_1$. Omitted.

∗ Case $(\texttt{RHFK})_1$.
1. $K_1 = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RHFK})_1$
2. $K_1' = \kappa_2', \kappa_3, \cdots, \kappa_n$ (IH), $(\texttt{RHFK})_1$
3. $(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa_2'$ (IH), $(\texttt{RHFK})_1$
4. $\underline{e_1} = \underline{e_1'}$ (IH), $(\texttt{RHFK})_1$
5. $\underline{e_1}\ \underline{e_2^n} = \underline{e_1'}\ \underline{e_2^n}$ (4)

6. If $|K_2| = |K_1|$ then
$\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n}\rangle$
(1), (2), (3), (5), $H \notin FH(K_2)$, $(\texttt{RHFK})_2$
7. If $|K_2| < |K_1|$ then
$\langle K_1 \bowtie K_2, \underline{e_1}\ \underline{e_2^n}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K_1' \bowtie K_2, \underline{e_1'}\ \underline{e_2^n}\rangle$
(1), (2), (3), (5), $H \notin FH(K_2)$, $(\texttt{RHFK})_1$

* Case $(\texttt{RHFK})_2$. Similar with $(\texttt{RHFK})_1$. Omitted.

- Case $(\texttt{APP}_\mathcal{S})_2$. Similar with $(\texttt{APP}_\mathcal{S})_1$. Omitted.
- Case $(\texttt{APP}_\mathcal{S})_3$. $(\lambda x.e^0)\ v^0 \xrightarrow{0} [x \mapsto v^0]e^0$.

1. $$\dfrac{\dfrac{r_0; x \vdash e^0 \mapsto (\underline{e^0}, \bot)}{r_0 \vdash \lambda x.e^0 \mapsto (\lambda x.\underline{e^0}, \bot)} \qquad r_0 \vdash v^0 \mapsto (\underline{v^0}, \bot)}{r_0 \vdash (\lambda x.e^0)\ v^0 \mapsto ((\lambda x.\underline{e^0})\ \underline{v_0}, \bot)}$$
2. $r_0 \setminus \{x\} \vdash e^0 \mapsto (\underline{e^0}, \bot)$ — (1), Lemma 5
3. $\bot \vdash v^0 \mapsto (\underline{v^0}, \bot)$ — (1), Lemma 5
4. $r_0 \setminus \{x\} \vdash [x \mapsto v^0]e^0 \mapsto (\{\underline{v^0}/x\}\underline{e^0}, \bot)$ — (2), (3), Lemma 8
5. $r_0 \vdash [x \mapsto v^0]e^0 \mapsto (\{\underline{v^0}/x\}\underline{e^0}, \bot)$ — (4), Lemma 5
6. $FH(\underline{e^0}) = FH(\underline{v^0}) = \varnothing$ — (1), Lemma 10
7. $(\lambda x.\underline{e^0})\ \underline{v_0} \xrightarrow{\mathcal{C}} \{\underline{v^0}/x\}\underline{e^0}$ — (6), $(\texttt{APP}_\mathcal{C})_3$
8. $(\lambda x.\underline{e^0})\ \underline{v_0} \xrightarrow{\mathcal{C};\mathcal{A}^*} \{\underline{v^0}/x\}\underline{e^0}$ — (7), Lemma 9
9. $\langle \bot, (\lambda x.\underline{e^0})\ \underline{v_0}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \{\underline{v^0}/x\}\underline{e^0}\rangle$ — (8), $(\texttt{RNOR})$

- Case $(\texttt{BOX}_\mathcal{S})$.

  – Case $n = 0$. $\dfrac{e \xrightarrow{1} e'}{\texttt{box}\ e \xrightarrow{0} \texttt{box}\ e'}$. By Lemma 1, $depth(e) = 1$. By Lemma 2, $|K| = depth(e) = 1$. Therefore $(\texttt{TBOX})_2$ cannot be applied, since $|K| = |\bot| = 0 \neq 1$. By $(\texttt{TBOX})_1$,

$$\dfrac{r_0, \bot \vdash e \mapsto (\underline{e}, \kappa_1)}{r_0 \vdash \texttt{box}\ e \mapsto (\kappa_1[\lambda u.\underline{e}], \bot)} \qquad \dfrac{r_0, \bot \vdash e' \mapsto (\underline{e'}, \kappa_1')}{r_0 \vdash \texttt{box}\ e' \mapsto (\kappa_1'[\lambda u'.\underline{e'}], \bot)}$$

By (IH), $\langle \kappa_1, \underline{e}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \kappa_1', \underline{e'}\rangle$. In this case, $(\texttt{RNOR})$, $(\texttt{RHFK})_1$ and $(\texttt{RHFK})_2$ cannot be applied.

* Case $(\texttt{RUNB})_1$.
1. $\kappa_1 = (\delta H.[\cdot]) \odot_\nu e_H$ — (IH), $(\texttt{RUNB})_1$
2. $\kappa_1' = (\delta H.[\cdot]) \odot_\nu e_H'$ — (IH), $(\texttt{RUNB})_1$
3. $\underline{e} = \underline{e'}$ — (IH), $(\texttt{RUNB})_1$
4. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e_H'$ — (IH), $(\texttt{RUNB})_1$
5.
$$\begin{aligned}
\kappa_1[\lambda u.\underline{e}] &= (\delta H.\lambda u.\underline{e})\ \odot_\nu\ e_H && (1)\\
&\xrightarrow{\mathcal{C};\mathcal{A}^*} (\delta H.\lambda u.\underline{e})\ \odot_\nu\ e_H' && ((4), \text{Lemma 9})\\
&= (\delta H.\lambda u.\underline{e'})\ \odot_\nu\ e_H' && (3)\\
&\equiv_\mathcal{C} (\delta H.\lambda u'.\underline{e'})\ \odot_\nu\ e_H' && (u' \notin FV(\underline{e'}))\\
&= \kappa_1'[\lambda u'.\underline{e'}] && (2)
\end{aligned}$$

6. $\langle \bot, \kappa_1[\lambda u.\underline{e}] \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \kappa_1'[\lambda u'.\underline{e'}] \rangle$ (5)

* Case $(\mathtt{RUNB})_2$. Similar with $(\mathtt{RUNB})_1$. Omitted.
* Case $(\mathtt{RHFE})_1$.

1. $\kappa_1 = (\delta H.[\cdot]) \odot_\nu e_H$ (IH), $(\mathtt{RHFE})_1$
2. $\kappa_1' = \bot$ (IH), $(\mathtt{RHFE})_1$
3. $(\underline{e}[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e'}$ (IH), $(\mathtt{RHFE})_1$
4. $e_H \in \mathit{Value}_{\mathcal{C}}$ (IH), $(\mathtt{RHFE})_1$

$$\begin{aligned}
\kappa_1[\lambda u.\underline{e}] &= (\delta H.\lambda u.\underline{e}) \;\odot_\nu\; e_H && (1)\\
&\xrightarrow{\mathcal{C}} ((\lambda u.\underline{e})[H^\nu/H])[e_H/H] && (4), ((\mathtt{HAPP}_{\mathcal{C}})_3)\\
&= \lambda u.(\underline{e}[H^\nu/H])[e_H/H] &&\\
&\xrightarrow{\mathcal{A}^*} \lambda u.\underline{e'} && (3)\\
&\equiv_{\mathcal{C}} \lambda u'.\underline{e'} && (u' \notin FV(\underline{e'}))\\
&= \bot[\lambda u'.\underline{e'}] &&\\
&= \kappa_1'[\lambda u'.\underline{e'}] && (2)
\end{aligned}$$

* Case $(\mathtt{RHFE})_2$.

1. $\kappa_1 = (\delta H.\kappa) \odot_\nu e_H$ (IH), $(\mathtt{RHFE})_1$
2. $\kappa_1' = \kappa$ (IH), $(\mathtt{RHFE})_1$
3. $(\underline{e}[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e'}$ (IH), $(\mathtt{RHFE})_1$
4. $e_H \in \mathit{Value}_{\mathcal{C}}$ (IH), $(\mathtt{RHFE})_1$
5. $H \notin FH(\kappa)$ (1), Lemma 3

$$\begin{aligned}
\kappa_1[\lambda u.\underline{e}] &= (\delta H.\kappa[\lambda u.\underline{e}]) \;\odot_\nu\; e_H && (1)\\
&\xrightarrow{\mathcal{C}} ((\kappa[\lambda u.\underline{e}])[H^\nu/H])[e_H/H] && (4), ((\mathtt{HAPP}_{\mathcal{C}})_3)\\
&= ((\kappa[H^\nu/H])[e_H/H])[\lambda u.(\underline{e}[H^\nu/H])[e_H/H]] &&\\
&= \kappa[\lambda u.(\underline{e}[H^\nu/H])[e_H/H]] && (5)\\
&\xrightarrow{\mathcal{A}^*} \kappa[\lambda u.\underline{e'}] && (3)\\
&\equiv_{\mathcal{C}} \kappa[\lambda u'.\underline{e'}] && (u' \notin FV(\underline{e'}))\\
&= \kappa_1'[\lambda u'.\underline{e'}] && (2)
\end{aligned}$$

– Case $n \geq 1$. $\dfrac{e \xrightarrow{n+1} e'}{\mathtt{box}\ e \xrightarrow{n} \mathtt{box}\ e'}$. $(\mathtt{TBOX})_2$ cannot be applied, by Lemma 1 and Lemma 2. By $(\mathtt{TBOX})_1$,

$$\frac{R, \bot \vdash e \mapsto (\underline{e}, (K, \kappa_{n+1}))}{R \vdash \mathtt{box}\ e \mapsto (\kappa_{n+1}[\lambda u.\underline{e}], K)} \qquad \frac{R, \bot \vdash e' \mapsto (\underline{e'}, (K', \kappa_{n+1}'))}{R \vdash \mathtt{box}\ e' \mapsto (\kappa_{n+1}'[\lambda u'.\underline{e'}], K')}$$

By (IH), $\langle (K, \kappa_{n+1}), \underline{e} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (K', \kappa_{n+1}'), \underline{e'} \rangle$. In this case, $(\mathtt{RNOR})$, $(\mathtt{RHFE})_1$ and $(\mathtt{RHFE})_2$ cannot be applied.

* Case $(\mathtt{RUNB})_1$.

1. $(K, \kappa_{n+1}) = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n, \kappa_{n+1}$ (IH), $(\mathtt{RUNB})_1$
2. $(K', \kappa_{n+1}') = (\delta H.[\cdot]) \odot_\nu e_H', \kappa_2, \cdots, \kappa_n, \kappa_{n+1}$ (1), (IH), $(\mathtt{RUNB})_1$
3. $\kappa_{n+1} = \kappa_{n+1}'$ (1), (2)

4. $K = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n$ (1)
5. $K' = (\delta H.[\cdot]) \odot_\nu e'_H, \kappa_2, \cdots, \kappa_n$ (2)
6. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RUNB})_1$
7. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H$ (IH), $(\texttt{RUNB})_1$
8.

$$\begin{aligned} \kappa_{n+1}[\lambda u.\underline{e}] &= \kappa'_{n+1}[\lambda u.\underline{e'}] && (3), (6)) \\ &\equiv_{\mathcal{C}} \kappa'_{n+1}[\lambda u'.\underline{e'}] && (u' \notin FV(\underline{e'})) \end{aligned}$$

9. $\langle K, \kappa_{n+1}[\lambda u.\underline{e}]\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \kappa'_{n+1}[\lambda u'.\underline{e'}]\rangle$ (4), (5), (7), (8), $(\texttt{RUNB})_1$

∗ Case $(\texttt{RUNB})_2$. Similar with $(\texttt{RUNB})_1$. Omitted.

∗ Case $(\texttt{RHFK})_1$.

· Case $n = 1$.

1. $(K, \kappa_2) = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2$ (IH), $(\texttt{RHFK})_1$
2. $(K', \kappa'_2) = \kappa'_2$ (IH), $(\texttt{RHFK})_1$
3. $K = (\delta H.[\cdot]) \odot_\nu e_H$ (1)
4. $K' = \bot$ (2)
5. $(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa'_2$ (IH), $(\texttt{RHFK})_1$
6. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RHFK})_1$
7. $e_H \in \mathit{Value}_C$ (IH), $(\texttt{RHFK})_1$
8. $H \notin \mathit{FH}(\underline{e})$ (1), Lemma 3
9.

$$\begin{aligned} ((\kappa_2[\lambda u.\underline{e}])[H^\nu/H])[e_H/H] &= ((\kappa_2[H^\nu/H])[e_H/H])[\lambda u.\underline{e}] && (8) \\ &\xrightarrow{\mathcal{C};\mathcal{A}^*} \kappa'_2[\lambda u.\underline{e}] && ((5), \text{Lemma 9}) \\ &= \kappa'_2[\lambda u.\underline{e'}] && (6) \\ &\equiv_{\mathcal{C}} \kappa'_2[\lambda u'.\underline{e'}] && (u' \notin FV(\underline{e'})) \end{aligned}$$

10. $\langle K, \kappa_2[\lambda u.\underline{e}]\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \kappa'_2[\lambda u'.\underline{e'}]\rangle$ (3), (4), (7), (9), $(\texttt{RHFE})_1$

· Case $n \geq 2$.

1. $(K, \kappa_{n+1}) = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n, \kappa_{n+1}$ (IH), $(\texttt{RHFK})_1$
2. $(K', \kappa'_{n+1}) = \kappa'_2, \kappa_3, \cdots, \kappa_n, \kappa_{n+1}$ (1), (IH), $(\texttt{RHFK})_1$
3. $\kappa_{n+1} = \kappa'_{n+1}$ (1), (2)
4. $K = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n$ (1)
5. $K' = \kappa'_2, \kappa_3, \cdots, \kappa_n$ (2)
6. $(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa'_2$ (IH), $(\texttt{RHFK})_1$
7. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RHFK})_1$
8.

$$\begin{aligned} \kappa_{n+1}[\lambda u.\underline{e}] &= \kappa'_{n+1}[\lambda u.\underline{e'}] && (3), (7) \\ &\equiv_{\mathcal{C}} \kappa'_{n+1}[\lambda u'.\underline{e'}] && (u' \notin FV(\underline{e'})) \end{aligned}$$

9. $\langle K, \kappa_{n+1}[\lambda u.\underline{e}]\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \kappa'_{n+1}[\lambda u'.\underline{e'}]\rangle$ (4), (5), (6), (8), $(\texttt{RHFK})_1$

∗ Case $(\texttt{RHFK})_2$. Similar with $(\texttt{RHFK})_1$. Omitted.

- Case $(\texttt{UNB}_{\mathcal{S}})_1$.

– Case $n = 1$. $\dfrac{e \xrightarrow{0} e'}{\texttt{unbox}\ e \xrightarrow{1} \texttt{unbox}\ e'}$.

$$\dfrac{r_0; r_1 \vdash e \mapsto (\underline{e}, \bot)}{r_0, r_1 \vdash \texttt{unbox}\ e \mapsto (H_1^{\nu^{-1}}\,(), (\delta H_1.[\cdot]) \odot_\nu \underline{e})} \quad \dfrac{r_0; r_1 \vdash e' \mapsto (\underline{e'}, \bot)}{r_0, r_1 \vdash \texttt{unbox}\ e' \mapsto (H_2^{\nu^{-1}}\,(), (\delta H_2.[\cdot]) \odot_\nu \underline{e'})}$$

1. $\langle \bot, \underline{e} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \underline{e'} \rangle$ (IH)
2. $\underline{e} \xrightarrow{\mathcal{C};\mathcal{A}^*} \underline{e'}$ (1), (RNOR)

$$\begin{aligned} \langle (\delta H_1.[\cdot]) \odot_\nu \underline{e}, H_1^{\nu^{-1}}\,() \rangle &\xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (\delta H_1.[\cdot]) \odot_\nu \underline{e'}, H_1^{\nu^{-1}}\,() \rangle && ((2), (\texttt{RUNB})_1) \\ &\equiv_{\mathcal{C}} \langle (\delta H_2.[\cdot]) \odot_\nu \underline{e'}, H_2^{\nu^{-1}}\,() \rangle \end{aligned}$$

– Case $n = 2$. $\dfrac{e \xrightarrow{1} e'}{\texttt{unbox}\ e \xrightarrow{2} \texttt{unbox}\ e'}$.

$$\dfrac{r_0, (r_1; r_2) \vdash e \mapsto (\underline{e}, K)}{r_0, r_1, r_2 \vdash \texttt{unbox}\ e \mapsto (H_1^{\nu^{-1}}\,(), (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}))} \quad \dfrac{r_0, (r_1; r_2) \vdash e' \mapsto (\underline{e'}, K')}{r_0, r_1, r_2 \vdash \texttt{unbox}\ e' \mapsto (H_2^{\nu^{-1}}\,(), (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}))}$$

By (IH), $\langle K, \underline{e} \rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \underline{e'} \rangle$. In this case, (RNOR), $(\texttt{RHFK})_1$ and $(\texttt{RHFK})_2$ cannot be applied.

∗ Case $(\texttt{RUNB})_1$.

1. $K = (\delta H.[\cdot]) \odot_\nu e_H$ (IH), $(\texttt{RUNB})_1$
2. $K' = (\delta H.[\cdot]) \odot_\nu e'_H$ (IH), $(\texttt{RUNB})_1$
3. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RUNB})_1$
4. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H$ (IH), $(\texttt{RUNB})_1$
5.

$$\begin{aligned} \langle (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}), H_1^{\nu^{-1}}\,() \rangle &\xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (K', (\delta H_1.[\cdot]) \odot_\nu \underline{e'}), H_1^{\nu^{-1}}\,() \rangle \\ & \qquad ((1), (2), (3), (4), (\texttt{RUNB})_1) \\ &\equiv_{\mathcal{C}} \langle (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}), H_2^{\nu^{-1}}\,() \rangle \end{aligned}$$

∗ Case $(\texttt{RUNB})_2$. Similar with $(\texttt{RUNB})_1$. Omitted.

∗ Case $(\texttt{RHFE})_1$.

1. $K = (\delta H.[\cdot]) \odot_\nu e_H$ (IH), $(\texttt{RHFE})_1$
2. $K' = \bot$ (IH), $(\texttt{RHFE})_1$
3. $(\underline{e}[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e'}$ (IH), $(\texttt{RHFE})_1$
4. $e_H \in \mathit{Value}_{\mathcal{C}}$ (IH), $(\texttt{RHFE})_1$
5.

$$\begin{aligned} (((\delta H_1.[\cdot]) \odot_\nu \underline{e})[H^\nu/H])[e_H/H] &= (\delta H_1.[\cdot]) \odot_\nu (\underline{e}[H^\nu/H])[e_H/H] \\ &\xrightarrow{\mathcal{A}^*} (\delta H_1.[\cdot]) \odot_\nu \underline{e'} && (3) \end{aligned}$$

$$\langle (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}), H_1^{\nu^{-1}}\,()\,\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (\delta H_1.[\cdot]) \odot_\nu \underline{e'}, H_1^{\nu^{-1}}\,()\,\rangle \quad ((1),\ (4),\ (5),\ (\mathtt{RHFK})_1)$$
$$= \langle (\bot, (\delta H_1.[\cdot]) \odot_\nu \underline{e'}), H_1^{\nu^{-1}}\,()\,\rangle$$
$$= \langle (K', (\delta H_1.[\cdot]) \odot_\nu \underline{e'}), H_1^{\nu^{-1}}\,()\,\rangle \quad (2)$$
$$\equiv_{\mathcal{C}} \langle (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}), H_2^{\nu^{-1}}\,()\,\rangle$$

  * Case $(\mathtt{RHFE})_2$. Similar with $(\mathtt{RHFE})_1$. Omitted.

- Case $n \geq 3$. $\dfrac{e \xrightarrow{n-1} e'}{\mathtt{unbox}\ e \xrightarrow{n} \mathtt{unbox}\ e'}$.

$$\cfrac{\cfrac{R, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, K)}{R, r_{n-1}, r_n \vdash \mathtt{unbox}\ e \mapsto (H_1^{\nu^{-1}}\,(), (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}))}}{} \qquad \cfrac{R, (r_{n-1}; r_n) \vdash e' \mapsto (\underline{e'}, K')}{R, r_{n-1}, r_n \vdash \mathtt{unbox}\ e' \mapsto (H_2^{\nu^{-1}}\,(), (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}))}$$

By (IH), $\langle K, \underline{e}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \underline{e'}\rangle$. In this case, $(\mathtt{RNOR})$, $(\mathtt{RHFE})_1$ and $(\mathtt{RHFE})_2$ cannot be applied.

  * Case $(\mathtt{RUNB})_1$.
    1. $K = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_{n-1}$ (IH), $(\mathtt{RUNB})_1$
    2. $K' = (\delta H.[\cdot]) \odot_\nu e'_H, \kappa_2, \cdots, \kappa_{n-1}$ (IH), $(\mathtt{RUNB})_1$
    3. $\underline{e} = \underline{e'}$ (IH), $(\mathtt{RUNB})_1$
    4. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H$ (IH), $(\mathtt{RUNB})_1$
    5.

$$\langle (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}), H_1^{\nu^{-1}}\,()\,\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (K', (\delta H_1.[\cdot]) \odot_\nu \underline{e'}), H_1^{\nu^{-1}}\,()\,\rangle \quad ((1),\ (2),\ (3),\ (4),\ (\mathtt{RUNB})_1)$$
$$\equiv_{\mathcal{C}} \langle (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}), H_2^{\nu^{-1}}\,()\,\rangle$$

  * Case $(\mathtt{RUNB})_2$. Similar with $(\mathtt{RUNB})_1$. Omitted.
  * Case $(\mathtt{RHFK})_1$.
    1. $K = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_{n-1}$ (IH), $(\mathtt{RHFK})_1$
    2. $K' = \kappa'_2, \kappa_3, \cdots, \kappa_{n-1}$ (IH), $(\mathtt{RHFK})_1$
    3. $(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa'_2$ (IH), $(\mathtt{RHFK})_1$
    4. $\underline{e} = \underline{e'}$ (IH), $(\mathtt{RHFK})_1$
    5.

$$\langle (K, (\delta H_1.[\cdot]) \odot_\nu \underline{e}), H_1^{\nu^{-1}}\,()\,\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle (K', (\delta H_1.[\cdot]) \odot_\nu \underline{e'}), H_1^{\nu^{-1}}\,()\,\rangle \quad ((1),\ (2),\ (3),\ (4),\ (\mathtt{RHFK})_1)$$
$$\equiv_{\mathcal{C}} \langle (K', (\delta H_2.[\cdot]) \odot_\nu \underline{e'}), H_2^{\nu^{-1}}\,()\,\rangle$$

  * Case $(\mathtt{RHFK})_2$. Similar with $(\mathtt{RHFK})_1$. Omitted.

- Case $(\mathtt{UNB}_{\mathcal{S}})_2$. $\mathtt{unbox}\ \left(\mathtt{box}\ v^1\right) \xrightarrow{1} v^1$.

1. $$\cfrac{\cfrac{(r_0; r_1), \bot \vdash v^1 \mapsto (\underline{v^1}, \bot)}{r_0; r_1 \vdash \mathtt{box}\ v^1 \mapsto (\lambda u.\underline{v^1}, \bot)}}{r_0, r_1 \vdash \mathtt{unbox}\ \left(\mathtt{box}\ v^1\right) \mapsto (H^{\nu^{-1}}\,(), (\delta H.[\cdot]) \odot_\nu \lambda u.\underline{v^1})}$$

2. $r_0, r_1 \vdash v^1 \mapsto (\underline{v^1}, \bot)$ (1), Lemma 5
3. $\lambda u.\underline{v^1} \in Value_{\mathcal{C}}$
4.

$$
\begin{aligned}
((H^{\nu}\,())[H^{\nu^{-1}}/H])[(\lambda u.\underline{v^1})/H] &= (H^{\nu^{-1}\star\nu}\,())[(\lambda u.\underline{v^1})/H] \\
&= (\overline{\nu^{-1}\star\nu}\ \lambda u.\underline{v^1})\ () \\
&= (\overline{\{\}}\ \lambda u.\underline{v^1})\ () \\
&= \lambda u.\underline{v^1}\ () \\
&\xrightarrow{\mathcal{A}} \{()/u\}\underline{v^1} \\
&= \underline{v^1} \qquad \text{(Remark 2)}
\end{aligned}
$$

5. $\langle(\delta H.[\cdot]) \odot_{\nu} \lambda u.\underline{v^1}, H^{\nu^{-1}}\,()\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle\bot, \underline{v^1}\rangle$ (3), (4), $(\texttt{RHFE})_1$

- Case $(\texttt{RUN}_{\mathcal{S}})_1$.

  – Case $n = 0$. $\dfrac{e \xrightarrow{0} e'}{\texttt{run}\ e \xrightarrow{0} \texttt{run}\ e'}$.

$$
\frac{r_0 \vdash e \mapsto (\underline{e}, \bot)}{r_0 \vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\,()), \bot)} \qquad \frac{r_0 \vdash e' \mapsto (\underline{e'}, \bot)}{r_0 \vdash \texttt{run}\ e' \mapsto (\texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\,()), \bot)}
$$

  1. $\langle\bot, \underline{e}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle\bot, \underline{e'}\rangle$ (IH)
  2. $\underline{e} \xrightarrow{\mathcal{C};\mathcal{A}^*} \underline{e'}$ (1), (RNOR)
  3.

$$
\begin{aligned}
\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\,()) &\xrightarrow{\mathcal{C};\mathcal{A}^*} \texttt{let}\ h = \underline{e'}\ \texttt{in}\ (h\,()) \qquad (2) \\
&\equiv_{\mathcal{C}} \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\,())
\end{aligned}
$$

  4. $\langle\bot, \texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\,())\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle\bot, \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\,())\rangle$ (3), (RNOR)

  – Case $n \geq 1$. $\dfrac{e \xrightarrow{n} e'}{\texttt{run}\ e \xrightarrow{n} \texttt{run}\ e'}$.

$$
\frac{R \vdash e \mapsto (\underline{e}, K)}{R \vdash \texttt{run}\ e \mapsto (\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\,()), K)} \qquad \frac{R \vdash e' \mapsto (\underline{e'}, K')}{R \vdash \texttt{run}\ e' \mapsto (\texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\,()), K')}
$$

  By (IH), $\langle K, \underline{e}\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \underline{e'}\rangle$. In this case, (RNOR)cannot be applied.

  * Case $(\texttt{RUNB})_1$.
    1. $K = (\delta H.[\cdot]) \odot_{\nu} e_H, \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RUNB})_1$
    2. $K' = (\delta H.[\cdot]) \odot_{\nu} e'_H, \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RUNB})_1$
    3. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RUNB})_1$
    4.

$$
\begin{aligned}
\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\,()) &= \texttt{let}\ h = \underline{e'}\ \texttt{in}\ (h\,()) \qquad (3) \\
&\equiv^{\mathsf{c}}_{\mathcal{C}} \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\,())
\end{aligned}
$$

    5. $e_H \xrightarrow{\mathcal{C};\mathcal{A}^*} e'_H$ (IH), $(\texttt{RUNB})_1$

6. $\langle K, \texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()})\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\texttt{()})\rangle$ (1), (2), (4), (5), $(\texttt{RUNB})_1$

- Case $(\texttt{RUNB})_2$. Similar with $(\texttt{RUNB})_1$. Omitted.
- Case $(\texttt{RHFE})_1$.

1. $K = (\delta H.[\cdot]) \odot_\nu e_H$ (IH), $(\texttt{RHFE})_1$
2. $K' = \bot$ (IH), $(\texttt{RHFE})_1$
3. $(\underline{e}[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \underline{e'}$ (IH), $(\texttt{RHFE})_1$
4.

$$\begin{aligned}((\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()}))[H^\nu/H])[e_H/H] &= \texttt{let}\ h = (\underline{e}[H^\nu/H])[e_H/H]\ \texttt{in}\ (h\texttt{()})\\ &\xrightarrow{\mathcal{A}^*} \texttt{let}\ h = \underline{e'}\ \texttt{in}\ (h\texttt{()}) && (3)\\ &\equiv_{\mathcal{C}} \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\texttt{()})\end{aligned}$$

5. $\langle K, \texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()})\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\texttt{()})\rangle$ (1), (2), (4), $(\texttt{RHFE})_1$

- Case $(\texttt{RHFE})_2$. Similar with $(\texttt{RHFE})_1$. Omitted.
- Case $(\texttt{RHFK})_1$.

1. $K = (\delta H.[\cdot]) \odot_\nu e_H, \kappa_2, \cdots, \kappa_n$ (IH), $(\texttt{RHFK})_1$
2. $K' = \kappa_2', \kappa_3, \cdots, \kappa_n$ (IH), $(\texttt{RHFK})_1$
3. $(\kappa_2[H^\nu/H])[e_H/H] \xrightarrow{\mathcal{A}^*} \kappa_2'$ (IH), $(\texttt{RHFK})_1$
4. $\underline{e} = \underline{e'}$ (IH), $(\texttt{RHFK})_1$
5.

$$\begin{aligned}\texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()}) &= \texttt{let}\ h = \underline{e'}\ \texttt{in}\ (h\texttt{()}) && (4)\\ &\equiv^{c}_{\mathcal{C}} \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\texttt{()})\end{aligned}$$

6. $\langle K, \texttt{let}\ h = \underline{e}\ \texttt{in}\ (h\texttt{()})\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle K', \texttt{let}\ h' = \underline{e'}\ \texttt{in}\ (h'\texttt{()})\rangle$ (1), (2), (3), (5), $(\texttt{RHFK})_1$

- Case $(\texttt{RHFK})_2$. Similar with $(\texttt{RHFK})_1$. Omitted.

- Case $(\texttt{RUN}_{\mathcal{S}})_2$. $\texttt{run}\ \left(\texttt{box}\ v^1\right) \xrightarrow{0} v^1$.

1. $$\cfrac{\cfrac{r_0, \bot \vdash v^1 \mapsto (\underline{v^1}, \bot)}{r_0 \vdash \texttt{box}\ v^1 \mapsto (\lambda u.\underline{v^1}, \bot)}}{r_0 \vdash \texttt{run}\ \left(\texttt{box}\ v^1\right) \mapsto (\texttt{let}\ h = \lambda u.\underline{v^1}\ \texttt{in}\ (h\texttt{()}), \bot)}$$
2. $r_0 \vdash v^1 \mapsto (\underline{v^1}, \bot)$ (1), Lemma 5
3. $\lambda u.\underline{v^1} \in \mathit{Value}_{\mathcal{C}}$
4.

$$\begin{aligned}\texttt{let}\ h = \lambda u.\underline{v^1}\ \texttt{in}\ (h\texttt{()}) &\xrightarrow{\mathcal{C}} \lambda u.\underline{v^1}\ \texttt{()} && (3)\\ &\xrightarrow{\mathcal{A}} \{\texttt{()}/u\}\underline{v^1}\\ &= \underline{v^1} && \text{(Lemma 2)}\end{aligned}$$

5. $\langle \bot, \texttt{let}\ h = \lambda u.\underline{v^1}\ \texttt{in}\ (h\texttt{()})\rangle \xrightarrow{\mathcal{C};\mathcal{A}^*} \langle \bot, \underline{v^1}\rangle$ (4), $(\texttt{RNOR})$

- Case $(\texttt{ABS}_{\mathcal{S}})$. Similar with $(\texttt{RUN}_{\mathcal{S}})_1$. Omitted.

□

Proof of Theorem 1. There exists such $\underline{e}$ and $\underline{e'}$ by Lemma 4, Corollary 1. The case for reducible expressions is a corollary of Lemma 11, and Lemma 6 proves the case for values. □

5.7 Inversion

Lemma 12. *Assume $\underline{e}$ is a $\lambda_{\mathcal{C}}$ expression, κ is a continuation, and $\mathcal{H}$ is a hole environment. If $\mathcal{H} \cup \overline{\kappa} \vdash \underline{e} \rightarrowtail e$ then $\mathcal{H} \vdash \kappa[\underline{e}] \rightarrowtail e$.*

Proof. by induction on κ.

- Let $\kappa = (\delta H.[\cdot]) \odot_\nu \underline{e_H}$.
  Suppose $\mathcal{H} \cup \overline{(\delta H.[\cdot]) \odot_\nu \underline{e_H}} \vdash \underline{e} \rightarrowtail e$. We have $\mathcal{H} \cup \{H : \underline{e_H}\} \vdash \underline{e} \rightarrowtail e$ since $\overline{(\delta H.[\cdot]) \odot_\nu \underline{e_H}} = \{H : \underline{e_H}\}$. Also, by definition of the inverse translation, we have $\mathcal{H} \vdash (\delta H.\underline{e}) \odot_\nu \underline{e_H} \rightarrowtail e$.

- Let $\kappa = (\delta H.\kappa') \odot_\nu \underline{e_H}$.
  Suppose $\mathcal{H} \cup \overline{(\delta H.\kappa') \odot_\nu \underline{e_H}} \vdash \underline{e} \rightarrowtail e$. We have $\mathcal{H} \cup \overline{\kappa'} \cup \{H : \underline{e_H}\} \vdash \underline{e} \rightarrowtail e$ since $\overline{(\delta H.\kappa') \odot_\nu \underline{e_H}} = \overline{\kappa'} \cup \{H : \underline{e_H}\}$. By induction hypothesis, $\mathcal{H} \cup \{H : \underline{e_H}\} \vdash \kappa'[\underline{e}] \rightarrowtail e$. Also, by definition of the inverse translation, we have $\mathcal{H} \vdash (\delta H.\kappa'[\underline{e}]) \odot_\nu \underline{e_H} \rightarrowtail e$.

□

Proof of Theorem 2. by structural induction on e. We are going to show only the interesting cases; other cases follow straightforward from the induction hypothesis.

- Let $e = x$.
  Suppose $R \vdash x \mapsto (x, \bot)$. We have $\varnothing \vdash x \rightarrowtail x$.

- Let $e = \texttt{box}\ e'$.
  Suppose $R, \bot \vdash e \mapsto (\underline{e}, K)$. We have two cases on translation of $\texttt{box}\ e$, depending on whether K equals to $\bot$ of not.

  – Case $K = \bot$.
    We have $R \vdash \texttt{box}\ e \mapsto (\lambda u.\underline{e}, \bot)$. By induction hypothesis, we have $\mathcal{H} \vdash \underline{e} \rightarrowtail e$ for any $\mathcal{H}$. So we have $\mathcal{H} \vdash \lambda u.\underline{e} \rightarrowtail \texttt{box}\ e$.

  – Case $K = K', \kappa$.
    We have $R \vdash \texttt{box}\ e \mapsto (\kappa[\lambda u.\underline{e}], K')$. Suppose $\overline{K'} \subseteq \mathcal{H}$. We have $\overline{K', \kappa} = \overline{K'} \cup \overline{\kappa} \subseteq \mathcal{H} \cup \overline{\kappa}$. By induction hypothesis, we have $\mathcal{H} \cup \overline{\kappa} \vdash \underline{e} \rightarrowtail e$. So we have $\mathcal{H} \cup \overline{\kappa} \vdash \lambda u.\underline{e} \rightarrowtail \texttt{box}\ e$. By Lemma 12, $\mathcal{H} \vdash \kappa[\lambda u.\underline{e}] \rightarrowtail \texttt{box}\ e$.

- Let $e = \texttt{unbox}\ e'$.
  Suppose $R, r_{n-1}, r_n \vdash \texttt{unbox}\ e \mapsto (H^{\nu^{-1}}\ (), (K, ((\delta H.[\cdot]) \odot_\nu \underline{e})))$. We have $R, (r_{n-1}; r_n) \vdash e \mapsto (\underline{e}, K)$, and $\nu = \{x/x \mid x \in r_n\}$. Suppose $\overline{(K, ((\delta H.[\cdot]) \odot_\nu \underline{e}))} \subseteq \mathcal{H}$. We have $\overline{(K, ((\delta H.[\cdot]) \odot_\nu \underline{e}))} = \overline{K} \cup \{H : \underline{e}\}$. Then, we have $\overline{K} \subseteq \overline{(K, ((\delta H.[\cdot]) \odot_\nu \underline{e}))} \subseteq \mathcal{H}$. By induction hypothesis, we have $\mathcal{H} \vdash \underline{e} \rightarrowtail e$. Also, $\mathcal{H}(H) = \underline{e}$. So we have $\mathcal{H} \vdash H^{\nu^{-1}}\ () \rightarrowtail \texttt{unbox}\ e$.

□

A Static Analysis of the Context Calculus

In this section, we present the set-based analysis [3] for the Context Calculus [2].

A.1 Construction

Figure 12 represents how set constraints are constructed in the context calculus. (`SCHVAR`), (`SCHAPP`), (`SCHABS`), and (`SCHUNIT`) are added to handle new terms of the context calculus. This expansion is natural because of the following reasons: (1) New terms such as $\delta H.e$ and $e_1 \odot_\nu e_2$ have a similar reduction to $\lambda x.e$ and $e_1\ e_2$. (2) Difference between the normal abstraction/application and the hole abstraction/application does not affect the construction of set constraints, since the set-based analysis assumes that all variables are distinct from each other.

A.2 Simplification

Simplification algorithm for the context calculus is also presented in Figure 12. Only case of *happly* added to usual simplification algorithm, and it is similar to the case of *apply*.

Construction of Set Constraints

(SCCONST) $$\frac{}{i \triangleright (\mathcal{X}, \{\mathcal{X} \supseteq i\})}$$

(SCVAR) $$\frac{}{x \triangleright (\mathcal{X}_x, \{\})}$$

(SCHVAR) $$\frac{}{H^\nu \triangleright (\mathcal{X}_H, \{\})}$$

(SCAPP) $$\frac{e_1 \triangleright (\mathcal{X}_1, \mathcal{C}_1) \quad e_2 \triangleright (\mathcal{X}_2, \mathcal{C}_2)}{e_1\ e_2 \triangleright (\mathcal{Y}, \{\mathcal{Y} \supseteq \mathit{apply}(\mathcal{X}_1, \mathcal{X}_2)\} \cup \mathcal{C}_1 \cup \mathcal{C}_2)}$$

(SCABS) $$\frac{e \triangleright (\mathcal{X}, \mathcal{C})}{\lambda x.e \triangleright (\mathcal{Y}, \{\mathcal{Y} \supseteq \lambda x.e, \mathit{ran}(\lambda x.e) \supseteq \mathcal{X}\} \cup \mathcal{C})}$$

(SCHAPP) $$\frac{e_1 \triangleright (\mathcal{X}_1, \mathcal{C}_1) \quad e_2 \triangleright (\mathcal{X}_2, \mathcal{C}_2)}{e_1 \odot_\nu e_2 \triangleright (\mathcal{Y}, \{\mathcal{Y} \supseteq \mathit{happly}(\mathcal{X}_1, \mathcal{X}_2)\} \cup \mathcal{C}_1 \cup \mathcal{C}_2)}$$

(SCHABS) $$\frac{e \triangleright (\mathcal{X}, \mathcal{C})}{\delta H.e \triangleright (\mathcal{Y}, \{\mathcal{Y} \supseteq \delta H.e, \mathit{hran}(\delta H.e) \supseteq \mathcal{X}\} \cup \mathcal{C})}$$

(SCHUNIT) $$\frac{}{() \triangleright (\mathcal{X}, \{\mathcal{X} \supseteq ()\})}$$

Simplification Algorithm

input a collection $\mathcal{C}$ of set constraints;
repeat
— **if** $\mathcal{X} \supseteq \mathit{apply}(\mathcal{X}_1, \mathcal{X}_2)$ and $\mathcal{X}_1 \supseteq \lambda x.e$ both appear in $\mathcal{C}$ **then**
— — **add** $\mathcal{X} \supseteq \mathit{ran}(\lambda x.e)$ to $\mathcal{C}$;
— — **add** $\mathcal{X}_x \supseteq \mathcal{X}_2$ to $\mathcal{C}$;
— **if** $\mathcal{X} \supseteq \mathit{happly}(\mathcal{X}_1, \mathcal{X}_2)$ and $\mathcal{X}_1 \supseteq \delta H.e$ both appear in $\mathcal{C}$ **then**
— — **add** $\mathcal{X} \supseteq \mathit{hran}(\delta H.e)$ to $\mathcal{C}$;
— — **add** $\mathcal{X}_x \supseteq \mathcal{X}_2$ to $\mathcal{C}$;
— **if** $\mathcal{X} \supseteq \mathcal{X}'$ and $\mathcal{X}' \supseteq ae$ both appear in $\mathcal{C}$,
— — where ae is atomic and not a set variable, **then**
— — **add** $\mathcal{X} \supseteq ae$ to $\mathcal{C}$;
until no step changes $\mathcal{C}$;
output $\mathit{explicit}(\mathcal{C})$;

Figure 12: Set-Based Analysis of the Context Calculus

References

[1] Wontae Choi, Baris Aktemur, Kwangkeun Yi, and Makoto Tatsuta. Static analysis of multi-staged programs via unstaging translation. In *Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '11, pages 81–92, New York, NY, USA, 2011. ACM.

[2] Masatomo Hashimoto and Atsushi Ohori. A typed context calculus. *Theor. Comput. Sci.*, 266(1-2):249–272, September 2001.

[3] Nevin Heintze. Set-based analysis of ml programs. In *Proceedings of the 1994 ACM conference on LISP and functional programming*, LFP '94, pages 306–317, New York, NY, USA, 1994. ACM.