Faithfulness of the VLISP Operational Semantics

William M. Farmer   Joshua D. Guttman   Leonard G. Monk
John D. Ramsdell   Vipin Swarup

The MITRE Corporation[1]
M92B093
September 1992

[1] This work was supported by Rome Laboratories of the United States Air Force, contract No. F19628-89-C-0001.
Authors' address: The MITRE Corporation, 202 Burlington Road, Bedford MA, 01730-1420.

Abstract

The Verified Programming Language Implementation project has developed a formally verified implementation of the Scheme programming language. This report provides a detailed proof that the operational semantics for the highest-level interpreter (virtual machine) is faithful to the denotational semantics used in the byte-code compiler proof.

Contents

1 Introduction

The VLISP project employs two kinds of semantics: one denotational and the other operational. This paper shows that the operational semantics is faithful to the denotational semantics. (The precise definition of faithfulness is given in section 4.) The two semantics are compared at the Tabular Byte Code (TBC) level, since TBC is assigned both kinds of semantics. (TBC is defined in [2].) The denotational semantics for TBC is presented as a set of equations in [1], and the operational semantics for TBC is defined by a state machine in [2]. This paper assumes familiarity with both [1] and [2].

The Augmented Byte Code (ABC) is an expansion of TBC defined in [2]. An ABC *state* is a finite sequence Σ of the form

$$\langle t, b, v, a, u, k, s \rangle,$$

with the abuse of notation that b is allowed to be $\langle\rangle$. We assign each ABC state a denotational answer $a \in \mathtt{A}$ such that $a : \mathtt{R}$ or $a = \bot_{\mathtt{A}}$. We also assign each ABC state an operational answer $a' \in \mathtt{A}$ with $a' : \mathtt{R}$, provided the state has a correctly terminating program run (in the operational semantics). Faithfulness is defined in terms of the denotational and operational answers assigned to well-structured ABC states—so-called "normal" states.

We will not consider the two semantics in detail for vectors and strings since their semantics is not essentially different from the semantics for pairs.

2 Denotational Answers

We define in this section the notion of a denotational answer of an ABC state, using some of the (denotational) semantic functions for TBC given in [1]. Our definition requires the following new semantic functions:

$\mathcal{D}_v : v \to \mathtt{E}$
$\mathcal{D}_a : v^* \to \mathtt{E}^*$
$\mathcal{D}_u : u \to \mathtt{N} \to \mathtt{N} \to \mathtt{L}$
$\mathcal{D}_k : k \to \mathtt{E} \to \mathtt{C}$
$\mathcal{D}_s : v^* \to \mathtt{E}^*$

The definitions of the functions take two parameters:

(1) a function $\rho_G : \mathrm{Ide} \to \mathtt{L}$ known as *globals*, and

(2) a function $\mathcal{K}' : \text{Con} \to \mathtt{E}$ known as *immutables* such that $\mathcal{K}' \subseteq \mathcal{K}$, $\mathcal{K}'(p)$ is defined only if p has the form $\langle \mathtt{immutable\text{-}pair}\ c_1\ c_2 \rangle$, and $\mathcal{K}'(p) : \mathtt{L} \times \mathtt{L} \times \{\mathit{immutable}\}$ if $\mathcal{K}'(p)$ is defined.

Also, for $f : A \to B$, $f^{(*)} : A^* \to B^*$ is defined by

$$f^{(*)}(\langle a_1, \ldots, a_n \rangle) = \langle f(a_1), \ldots, f(a_n) \rangle,$$

and $f^{[*]} : A^* \to B^*$ is defined by

$$f^{[*]}(\langle a_1, \ldots, a_n \rangle) = \langle f(a_n), \ldots, f(a_1) \rangle.$$

The semantic functions are defined by the following equations:

$\mathcal{D}_v[\![c]\!] = \mathcal{K}[\![c]\!]$.
$\mathcal{D}_v[\![\langle \text{CLOSURE}\ t\ u\ l \rangle]\!] = \mathit{fix}(\lambda\epsilon.\langle l, (\lambda\epsilon^*\kappa.\mathcal{T}_\tau[\![t]\!]\rho_G\epsilon\epsilon^*\mathcal{D}_u[\![u]\!](\lambda\epsilon.\kappa\langle\epsilon\rangle))\rangle)$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\langle \text{ESCAPE}\ k\ l \rangle]\!] = \langle l, \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![k]\!]\epsilon)\rangle$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle]\!] = \langle l_1, l_2, \mathit{mutable}\rangle$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\langle \text{STRING}\ l^* \rangle]\!] = \langle l^*, \mathit{mutable}\rangle$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\langle \text{VECTOR}\ l^* \rangle]\!] = \langle l^*, \mathit{mutable}\rangle$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\text{NOT-SPECIFIED}]\!] = \mathit{unspecified}$ in $\mathtt{E}$.
$\mathcal{D}_v[\![\text{UNDEFINED}]\!] = \mathit{empty}$ in $\mathtt{E}$.

$\mathcal{D}_a = \mathcal{D}_v^{[*]}$.

$\mathcal{D}_u[\![\text{EMPTY-ENV}]\!] = (\lambda\nu_1\nu_2.\bot)$.
$\mathcal{D}_u[\![\langle \text{ENV}\ u\ l^* \rangle]\!] = \mathit{extend}_R\,\mathcal{D}_u[\![u]\!](\mathit{rev}\ l^*)$.

$\mathcal{D}_k[\![\text{HALT}]\!] = (\lambda\epsilon\sigma.\epsilon : \mathtt{R} \to \epsilon|\mathtt{R}\ \text{in}\ \mathtt{A}, \bot)$.
$\mathcal{D}_k[\![\langle \text{CONT}\ \langle \mathtt{template}\ b_0\ e \rangle\ b\ a\ u\ k \rangle]\!] = \lambda\epsilon.\mathcal{B}_\tau[\![b]\!]e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]$.

$\mathcal{D}_s = \mathcal{D}_v^{(*)}$.

Note that the following identities hold:

(1) $\mathcal{T}_\tau[\![\langle \mathtt{template}\ b\ e \rangle]\!] = \mathcal{B}_\tau[\![b]\!]e$.

(2) $\mathcal{D}_v[\![\langle \mathtt{immutable\text{-}pair}\ c_1\ c_2 \rangle]\!] = \mathcal{K}'[\![\langle \mathtt{immutable\text{-}pair}\ c_1\ c_2 \rangle]\!]$.

The *denotational answer* of an ABC state $\Sigma = \langle t, b, v, a, u, k, s \rangle$, denoted $\mathcal{D}[\![\Sigma]\!]$, is the value of

$$\mathcal{B}_\tau[\![b]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!].$$

3 Operational Answers

The operational semantics for TBC is defined by the Tabular Byte Code State Machine presented in [2]. The machine actions are specified by a set of rules called ABC rules. We define in this section the notion of a operational answer of an ABC state, using the ABC rules.

The *operational answer* of an ABC state Σ, denoted $\mathcal{O}[\![\Sigma]\!]$, is defined inductively by:

(1) If $b = \langle\rangle$ and $v \in \mathtt{R}$, then $\mathcal{O}[\![\Sigma]\!] = \mathcal{D}_v[\![v]\!]|\mathtt{R}$ in $\mathtt{A}$.

(2) If $b = \langle\rangle$ and $v \notin \mathtt{R}$, then $\mathcal{O}[\![\Sigma]\!]$ is undefined.

(3) If $b \neq \langle\rangle$, then $\mathcal{O}[\![\Sigma]\!]$ is the unique $a \in \mathtt{A}$ for which there is an ABC rule R such that $a = \mathcal{O}[\![R(\Sigma)]\!]$ (and is undefined if there is no such unique a).

Note that, by the definition of $\mathcal{K}$ on $\mathtt{R}$ (see [1]), if $v \in \mathtt{R}$, then $\mathcal{D}_v[\![v]\!] : \mathtt{R}$.

$\mathcal{O}'[\![\Sigma]\!]$ is defined exactly like $\mathcal{O}[\![\Sigma]\!]$ except that the following rule is used in place of the Make Environment ABC rule:

Rule: **Alternate Make Environment**
Domain Conditions:
$b = \langle \mathtt{make\text{-}env}\ \#a\rangle :: b_1$
Changes:
$b' = b_1$
$a' = \langle\rangle$
$u' = \mathit{add\text{-}layer}'(u,\ \#s,\ \#a)$
$s' = s \frown (\mathit{rev}\ a)$

where *add-layer'* is defined by:

$$\mathit{add\text{-}layer}'(u, n_0, n_1) = \begin{cases} \langle \text{ENV}\ u\ \langle (n_0 + n_1 - 1) \cdots n_0 \rangle\rangle & \text{if } n_1 > 0 \\ \langle \text{ENV}\ u\ \langle\rangle\rangle & \text{if } n_1 = 0 \end{cases}$$

The *special rules* are Alternate Make Environment and the ABC rules other than Make Environment.

4 Normal States and Faithfulness

Let $\Sigma = \langle\langle \texttt{template}\ b_0\ e\rangle, b, v, a, u, k, s\rangle$ be an ABC state. The functions $L_{\rm glo}, L_{\rm env}, L_{\rm mp}, L_{\rm ip}$, which map ABC states to finite sets of locations, are defined by the following equations:

$$L_{\rm glo}(\Sigma) = ran(\rho_G).$$

$$L_{\rm env}(\Sigma) = \{\textit{env-reference}(u', n_0, n_1) : u' \text{ occurs in } \Sigma,\ n_0, n_1 \in \mathtt{N}, \text{ and } \textit{env-reference}(u', n_0, n_1) \text{ is defined.}\}.$$

$$L_{\rm mp}(\Sigma) = \{l_1, l_2 \in \mathtt{L} : \langle \text{MUTABLE-PAIR}\ l_1\ l_2\rangle \text{ occurs in } \Sigma\}.$$

$$L_{\rm ip}(\Sigma) = \{l_1, l_2 \in \mathtt{L} : p = \langle \texttt{immutable-pair}\ c_1\ c_2\rangle \text{ occurs } \Sigma \text{ and } (\mathcal{K}'[\![p]\!] | \mathtt{L} \times \mathtt{L} \times \{\textit{immutable}\}) = \langle l_1, l_2, \textit{immutable}\rangle\}.$$

Σ is *normal* if the following conditions hold:

(1) $L_{\rm glo}(\Sigma) \cup L_{\rm env}(\Sigma) \cup L_{\rm ip}(\Sigma) \cup L_{\rm mp}(\Sigma) \subseteq dom(s)$.

(2) $L_{\rm glo}(\Sigma)$, $L_{\rm env}(\Sigma)$, $L_{\rm ip}(\Sigma)$, $L_{\rm mp}(\Sigma)$ are pairwise disjoint.

(3) For each $p = \langle \texttt{immutable-pair}\ c_1\ c_2\rangle$ occurring in Σ, there are $l_1, l_2 \in \mathtt{L}$ such that $(\mathcal{K}'[\![p]\!] | \mathtt{L} \times \mathtt{L} \times \{\textit{immutable}\}) = \langle l_1, l_2, \textit{immutable}\rangle$, $s(l_1) = c_1$, and $s(l_2) = c_2$.

A *initial state* for a TBC template $t = \langle \texttt{template}\ b\ e\rangle$ is any normal ABC state of the form

$$\langle t, b, \text{NOT-SPECIFIED}, \langle\rangle, \text{EMPTY-ENV}, \text{HALT}, s\rangle.$$

The operational semantics for TBC is *faithful* if, for all TBC templates t and all initial states Σ for t, $\mathcal{D}[\![\Sigma]\!] = \mathcal{O}[\![\Sigma]\!]$ whenever $\mathcal{O}[\![\Sigma]\!]$ is defined.

5 The Faithfulness Theorem

In this section, we shall prove that the operational semantics for TBC is faithful.

Lemma 5.1 *If Σ is a normal* ABC *state, R is an* ABC *or special rule, and $\Sigma' = R(\Sigma)$, then Σ' is also a normal* ABC *state.*

Proof Let $\Sigma = \langle t, b, v, a, u, k, s \rangle$ be a normal ABC state, R be an ABC or special rule, and $\Sigma' = \langle t', b', v', a', u', k', s' \rangle = R(\Sigma)$. Σ' is certainly normal if:

(1) s is a subfunction of s' and

(2) Σ and Σ' contain exactly the same environments, mutable pairs, and immutable pairs.

There are just seven ABC or special rules which cause (1) or (2) to fail to hold: Set Global, Set Local, Make Environment, Alternate Make Environment, Make Rest List, Primitive Cons, and Primitive Set-car!.

Set Global modifies the value in s at a location in L_{glo}, but creates no new environments, mutable pairs, or immutable pairs.

Set Local modifies the value in s at a location in L_{env}, but creates no new environments, mutable pairs, or immutable pairs.

Make Environment and Alternate Make Environment both extend the store s to s' and increases u to u'. However,

$$L_{\text{env}}(\Sigma') \setminus L_{\text{env}}(\Sigma) = \mathit{dom}(s') \setminus \mathit{dom}(s).$$

Also, Make Environment and Alternate Make Environment both create no new mutable pairs or immutable pairs and no other new environments.

Make Rest List extends the store s to s' and makes new mutable pairs. However,

$$L_{\text{mp}}(\Sigma') \setminus L_{\text{mp}}(\Sigma) \subseteq \mathit{dom}(s') \setminus \mathit{dom}(s).$$

Also, Make Rest List creates no new immutable pairs or environments.

Primitive Cons extends the store s to s' and sets v' to a new mutable pair $\langle \text{MUTABLE-PAIR}\, l_1\, l_2 \rangle$. However, $\mathit{dom}(s') \setminus \mathit{dom}(s) = \{l_1, l_2\}$. Also, Primitive Cons creates no new environments or immutable pairs and no other mutable pairs.

Primitive Set-car! modifies the value in s at a location in L_{mp}, but creates no new environments, mutable pairs, or immutable pairs.

Therefore, Σ' is normal when R is any one of these seven rules. □

Lemma 5.2 *If $\Sigma = \langle t, \langle\rangle, v, a, u, k, s \rangle$ is a normal* ABC *state with $v \in \mathtt{R}$, then $\mathcal{D}[\![\Sigma]\!] = \mathcal{D}_v[\![v]\!] | \mathtt{R}$ in $\mathtt{A}$.*

Proof Let $\Sigma = \langle\langle\texttt{template}\ b\ e\rangle, \langle\rangle, v, a, u, k, s\rangle$ be a normal ABC state with $v \in \texttt{R}$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_{\mathcal{T}}[\![\langle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\epsilon\epsilon^*\rho_R\psi\sigma.\epsilon : \texttt{R} \to \epsilon|\texttt{R}\text{ in }\texttt{A}, \bot) \\
&\quad\ e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}_v[\![v]\!]|\texttt{R}\text{ in }\texttt{A}
\end{aligned}$$

since $v \in \texttt{R}$ implies $\mathcal{D}_v[\![v]\!] : \texttt{R}$.□

The proof of the following lemma is postponed until section 6.

Lemma 5.3 *If Σ and Σ' are normal* ABC *states such that $\mathcal{O}'[\![\Sigma]\!]$ is defined and $\Sigma' = R(\Sigma)$ for some special rule R, then $\mathcal{D}[\![\Sigma]\!] = \mathcal{D}[\![\Sigma']\!]$.*

The proof of the next lemma is postponed until section 7.

Lemma 5.4 *For each* TBC *template t and initial state Σ for t, if $\mathcal{O}[\![\Sigma]\!]$ is defined, then $\mathcal{O}[\![\Sigma]\!] = \mathcal{O}'[\![\Sigma]\!]$.*

Theorem 5.5 *The operational semantics for* TBC *is faithful.*

Proof Let $t = \langle\texttt{template}\ b\ e\rangle$ be a TBC template, and let

$$\Sigma = \langle t, b, \text{NOT-SPECIFIED}, \langle\rangle, \text{EMPTY-ENV}, \text{HALT}, s\rangle$$

be an initial state for t. Assume $\mathcal{O}[\![\Sigma]\!] = a$, and so by Lemma 5.4, $\mathcal{O}'[\![\Sigma]\!] = a$. We shall show that $\mathcal{D}[\![\Sigma]\!] = a$.

Since $\mathcal{O}'[\![\Sigma]\!]$ is defined, there is a finite sequence $\Sigma_0, \ldots, \Sigma_n$ of ABC states and a finite sequence $R_1, \ldots, R_n$ of special rules such that:

(1) $\Sigma = \Sigma_0$ and $0 \leq n$.

(2) $R_{i+1}(\Sigma_i) = \Sigma_{i+1}$ for all i with $0 \leq i \leq n-1$.

(3) Σ_n has the form $\langle t', \langle\rangle, v, a, u, k, s'\rangle$ with $(\mathcal{D}_v[\![v]\!]|\texttt{R}\text{ in }\texttt{A}) = a$.

Σ_i is normal for all i with $1 \leq i \leq n$ by Lemma 5.1, and $\mathcal{O}'[\![\Sigma_i]\!] = a$ for all i with $0 \leq i \leq n$ by the definition of $\mathcal{O}'$. Thus $\mathcal{D}[\![\Sigma_n]\!] = a$ by Lemma 5.2, and $\mathcal{D}[\![\Sigma_i]\!] = \mathcal{D}[\![\Sigma_{i+1}]\!]$ for all i with $0 \leq i \leq n-1$ by Lemma 5.3. Therefore, $\mathcal{D}[\![\Sigma]\!] = a = \mathcal{O}[\![\Sigma]\!]$. □

6 Proof of Lemma 5.3

We use in the proof of Lemma 5.3 the following easily verified facts:

- $\#\mathcal{D}_a[\![a]\!] = \#a$.
- $\#\mathcal{D}_s[\![s]\!] = \#s$.
- $\mathcal{D}_v[\![v]\!] = \mathit{false}$ in $\mathtt{E}$ iff $v = \mathit{false}$.
- $\mathit{single}\,\psi\langle\epsilon\rangle = \psi\epsilon$ (and so $\lambda\epsilon.\mathit{single}\,\psi\langle\epsilon\rangle = \psi$).
- If $l \leq \#\sigma$, then $\mathit{update}\,l\epsilon\sigma = \sigma[\epsilon/l]$ (and so $\mathit{update}\,(\mathit{new}\,\sigma)\epsilon\sigma = \sigma[\epsilon/\#\sigma]$).

The proof of Lemma 5.3 breaks down into 34 cases, one for each special rule. Let Σ and Σ' be normal ABC states such that $\mathcal{O}'[\![\Sigma]\!]$ is defined and $\Sigma' = R(\Sigma)$ for some special rule R. To prove Lemma 5.3, we must show that, for each special rule R, if Σ satisfies the domain conditions of R, then $\mathcal{D}[\![\Sigma]\!] = \mathcal{D}[\![\Sigma']\!]$.

Case 1: $R =$ Return-Halt.

Let $\Sigma = \langle t, \langle\langle \mathtt{return}\rangle\rangle, v, a, u, \text{HALT}, s\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, \langle\rangle, v, a, u, \text{HALT}, s\rangle$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\langle \mathtt{return}\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{return})e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{return}\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon\epsilon^*\rho_R\psi.\psi\epsilon)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon\sigma.\epsilon : \mathtt{R} \to \epsilon|\mathtt{R} \text{ in } \mathtt{A}, \bot)\mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\epsilon\epsilon^*\rho_R\psi\sigma.\epsilon : \mathtt{R} \to \epsilon|\mathtt{R} \text{ in } \mathtt{A}, \bot) \\
&\quad\; e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![\langle\langle\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![\text{HALT}]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 2: $R =$ Return.

Let $\Sigma = \langle\langle \texttt{template}\ b\ e\rangle, \langle\langle \texttt{return}\rangle\rangle, v, a, u, k, s\rangle$,
where $k = \langle \textsc{cont}\ t_1\ b_1\ a_1\ u_1\ k_1\rangle$ and $t_1 = \langle \texttt{template}\ b'\ e_1\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, b_1, v, a_1, u_1, k_1, s\rangle$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\langle \texttt{return}\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= (\lambda e\rho.\mathit{return})e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= \mathit{return}\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= (\lambda\epsilon\epsilon^*\rho_R\psi.\psi\epsilon)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= \mathcal{D}_k[\![k]\!]\mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!]\\
&= (\lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\epsilon\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k_1]\!])\mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!]\\
&= \mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k_1]\!]\mathcal{D}_s[\![s]\!]\\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 3: $R =$ Call.

Let $\Sigma = \langle\langle \texttt{template}\ b\ e\rangle, \langle\langle \texttt{call}\ \#a\rangle\rangle, v, a, u, k, s\rangle$,
where $v = \langle \textsc{closure}\ t_1\ u_1\ l_1\rangle$ and $t_1 = \langle \texttt{template}\ b_1\ e_1\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, b_1, v, a, u_1, k, s\rangle$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\langle \texttt{call}\ \#a\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= (\lambda e\rho.\mathit{call}\ \#a)e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= \mathit{call}\ \#a\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= (\lambda\nu\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* = \nu \to \mathit{applicate}\ \epsilon\epsilon^*(\mathit{single}\ \psi),\\
&\quad\ \mathit{wrong}\ \text{“bad stack”})\#a\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\\
&= (\#\mathcal{D}_a[\![a]\!] = \#a \to \mathit{applicate}\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!]),\\
&\quad\ \mathit{wrong}\ \text{“bad stack”})\mathcal{D}_s[\![s]\!]\\
&= \mathit{applicate}\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!]\\
&= (\lambda\epsilon\epsilon^*\kappa.\epsilon : \mathtt{F} \to ((\epsilon|\mathtt{F})1)\epsilon^*\kappa, \mathit{wrong}\ \text{“bad procedure”})\\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!]\\
&= (\mathcal{D}_v[\![v]\!] : \mathtt{F} \to ((\mathcal{D}_v[\![v]\!]|\mathtt{F})1)\mathcal{D}_a[\![a]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!]),\\
&\quad\ \mathit{wrong}\ \text{“bad procedure”})\mathcal{D}_s[\![s]\!]\\
&= (\lambda\epsilon^*\kappa.\mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v]\!]\epsilon^*\mathcal{D}_u[\![u_1]\!](\lambda\epsilon.\kappa\langle\epsilon\rangle))
\end{aligned}$$

$$
\begin{aligned}
& \mathcal{D}_a[\![a]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
= \ & \mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u_1]\!](\lambda\epsilon.\mathit{single}\ \mathcal{D}_k[\![k]\!]\langle\epsilon\rangle)\mathcal{D}_s[\![s]\!] \\
= \ & \mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \ & \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 4: $R =$ Escape-Halt.

Let $\Sigma = \langle t, \langle\langle \texttt{call}\ 1\rangle\rangle, v, a, u, k, s\rangle$,
where $v = \langle \textsc{escape halt}\ l\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, \langle\rangle, v, a, u, k, s\rangle$.

$\mathcal{O}'[\![\Sigma']\!]$ is undefined since $v \notin \texttt{R}$. Thus $\mathcal{O}'[\![\Sigma]\!]$ is undefined, and so we do not have to consider this case.

Case 5: $R =$ Escape.

Let $\Sigma = \langle\langle \texttt{template}\ b\ e\rangle, \langle\langle \texttt{call}\ 1\rangle\rangle, v, \langle v_1\rangle, u, k, s\rangle$,
where $v = \langle \textsc{escape}\ k'\ l\rangle$, $k' = \langle \textsc{cont}\ t_1\ b_1\ a_1\ u_1\ k_1\rangle$, and $t_1 = \langle \texttt{template}\ b'\ e_1\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, b_1, v_1, a_1, u_1, k_1, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] \ &= \ \mathcal{B}_\tau[\![\langle\langle \texttt{call}\ 1\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \ (\lambda e\rho.\mathit{call}\ 1)e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \ \mathit{call}\ 1\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \ (\lambda\nu\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* = \nu \rightarrow \mathit{applicate}\ \epsilon\epsilon^*(\mathit{single}\ \psi), \\
&\qquad \mathit{wrong}\ \text{“bad stack”})\ 1\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \ (\#\mathcal{D}_a[\![\langle v_1\rangle]\!] = 1 \rightarrow \mathit{applicate}\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!]), \\
&\qquad \mathit{wrong}\ \text{“bad stack”})\mathcal{D}_s[\![s]\!] \\
&= \ \mathit{applicate}\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \ (\lambda\epsilon\epsilon^*\kappa.\epsilon : \texttt{F} \rightarrow ((\epsilon|\texttt{F})1)\epsilon^*\kappa, \mathit{wrong}\ \text{“bad procedure”}) \\
&\qquad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \ (\mathcal{D}_v[\![v]\!] : \texttt{F} \rightarrow ((\mathcal{D}_v[\![v]\!]|\texttt{F})1)\mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!]), \\
&\qquad \mathit{wrong}\ \text{“bad procedure”})\mathcal{D}_s[\![s]\!] \\
&= \ \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![k']\!]\epsilon)\mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!]
\end{aligned}
$$

$$
\begin{aligned}
&= (\lambda\epsilon\kappa.\mathcal{D}_k[\![k']\!]\epsilon)(\mathcal{D}_a[\![\langle v_1\rangle]\!]0)(\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}_k[\![k']\!]\mathcal{D}_v[\![v_1]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\epsilon\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k_1]\!])\mathcal{D}_v[\![v_1]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\mathcal{D}_v[\![v_1]\!]\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k_1]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 6: $R =$ Closed Branch/True.

Let $\Sigma = \langle t, \langle\langle \texttt{unless-false}\ b_1\ b_2\rangle\rangle, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $v \neq \mathit{false}$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v, a, u, k, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\langle \texttt{unless-false}\ b_1\ b_2\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{if\_truish}(\mathcal{B}_\tau[\![b_1]\!]e\rho)(\mathcal{B}_\tau[\![b_2]\!]e\rho))e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad \mathcal{D}_s[\![s]\!] \\
&= \mathit{if\_truish}(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathcal{B}_\tau[\![b_2]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\pi_1\pi_2\epsilon\epsilon^*\rho_R\psi.\mathit{truish}\ \epsilon \to \pi_1\epsilon\epsilon^*\rho_R\psi, \pi_2\epsilon\epsilon^*\rho_R\psi) \\
&\quad (\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathcal{B}_\tau[\![b_2]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 7: $R =$ Closed Branch/False.

Let $\Sigma = \langle t, \langle\langle \texttt{unless-false}\ b_1\ b_2\rangle\rangle, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $v = \mathit{false}$.

Then $R(\Sigma) = \Sigma' = \langle t, b_2, v, a, u, k, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\langle \texttt{unless-false}\ b_1\ b_2\rangle\rangle]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{if\_truish}(\mathcal{B}_\tau[\![b_1]\!]e\rho)(\mathcal{B}_\tau[\![b_2]\!]e\rho))e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad \mathcal{D}_s[\![s]\!] \\
&= \mathit{if\_truish}(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathcal{B}_\tau[\![b_2]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\pi_1\pi_2\epsilon\epsilon^*\rho_R\psi.\mathit{truish}\ \epsilon \to \pi_1\epsilon\epsilon^*\rho_R\psi, \pi_2\epsilon\epsilon^*\rho_R\psi) \\
&\quad (\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathcal{B}_\tau[\![b_2]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]
\end{aligned}
$$

$$
\begin{aligned}
&= (\mathcal{B}_\tau[\![b_2]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 8: $R =$ Open Branch/True.

Let $\Sigma = \langle t, \langle \texttt{unless-false}\ y_1\ y_2\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $v \neq \mathit{false}$.

Then $R(\Sigma) = \Sigma' = \langle t, y_1 {}^\smile b_1, v, a, u, k, s\rangle$.

First, we shall show that

$$
\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) = \mathcal{B}_\tau[\![y_1 {}^\smile b_1]\!]e\rho_G
$$

by induction on the structure of y_1. There are four cases depending on the form of y_1:

(1) Assume $y_1 = \langle z\rangle$. Then

$$
\begin{aligned}
\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) &= (\lambda e\rho\pi.\mathcal{Z}_\tau[\![z]\!]e\rho(\mathcal{Y}_\tau[\![\langle\rangle]\!]e\rho\pi)) \\
&\quad\ e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&= \mathcal{Z}_\tau[\![\langle z\rangle]\!]e\rho_G((\lambda e\rho\pi.\pi)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
&= \mathcal{Z}_\tau[\![\langle z\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle z\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&= \mathcal{B}_\tau[\![z :: b_1]\!]e\rho_G \\
&= \mathcal{B}_\tau[\![y_1 {}^\smile b_1]\!]e\rho_G
\end{aligned}
$$

(2) Assume $y_1 = z :: y'$. Then

$$
\begin{aligned}
\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) &= (\lambda e\rho\pi.\mathcal{Z}_\tau[\![z]\!]e\rho(\mathcal{Y}_\tau[\![y']\!]e\rho\pi)) \\
&\quad\ e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&= \mathcal{Z}_\tau[\![z]\!]e\rho_G(\mathcal{Y}_\tau[\![y']\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
&= \mathcal{Z}_\tau[\![z]\!]e\rho_G(\mathcal{B}_\tau[\![y' {}^\smile b_1]\!]e\rho_G) \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![z]\!]e\rho(\mathcal{B}_\tau[\![y' {}^\smile b_1]\!]e\rho))e\rho_G \\
&= \mathcal{B}_\tau[\![z :: y' {}^\smile b_1]\!]e\rho_G \\
&= \mathcal{B}_\tau[\![y_1 {}^\smile b_1]\!]e\rho_G
\end{aligned}
$$

(3) Assume $y_1 = \langle \texttt{make-cont}\ \langle\rangle\ n\rangle :: b'$. Then

$$\begin{aligned}
\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) &= (\lambda e\rho\pi.\mathit{make_cont}(\mathcal{Y}_\tau[\![\langle\rangle]\!]e\rho\pi)n(\mathcal{B}_\tau[\![b']\!]e\rho)) \\
&\quad e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&= \mathit{make_cont}(\mathcal{Y}_\tau[\![\langle\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
&\quad n(\mathcal{B}_\tau[\![b']\!]e\rho_G) \\
&= \mathit{make_cont}((\lambda e\rho\pi.\pi)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
&\quad n(\mathcal{B}_\tau[\![b']\!]e\rho_G) \\
&= \mathit{make_cont}(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)n(\mathcal{B}_\tau[\![b']\!]e\rho_G) \\
&= (\lambda e\rho.\mathit{make_cont}(\mathcal{B}_\tau[\![b_1]\!]e\rho)n(\mathcal{B}_\tau[\![b']\!]e\rho))e\rho_G \\
&= \mathcal{B}_\tau[\![\langle \texttt{make-cont}\ b_1\ n\rangle :: b']\!]e\rho_G \\
&= \mathcal{B}_\tau[\![y_1{}^\smile b_1]\!]e\rho_G
\end{aligned}$$

(4) Assume $y_1 = \langle \texttt{make-cont}\ y'\ n\rangle :: b'$. Then

$$\begin{aligned}
\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) &= (\lambda e\rho\pi.\mathit{make_cont}(\mathcal{Y}_\tau[\![y']\!]e\rho\pi)n(\mathcal{B}_\tau[\![b']\!]e\rho)) \\
&\quad e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&= \mathit{make_cont}(\mathcal{Y}_\tau[\![y']\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
&\quad n(\mathcal{B}_\tau[\![b']\!]e\rho_G) \\
&= \mathit{make_cont}(\mathcal{B}_\tau[\![y'{}^\smile b_1]\!]e\rho_G)n(\mathcal{B}_\tau[\![b']\!]e\rho_G) \\
&= (\lambda e\rho.\mathit{make_cont}(\mathcal{B}_\tau[\![y'{}^\smile b_1]\!]e\rho)n(\mathcal{B}_\tau[\![b']\!]e\rho)) \\
&\quad e\rho_G \\
&= \mathcal{B}_\tau[\![\langle \texttt{make-cont}\ y'{}^\smile b_1\ n\rangle :: b']\!]e\rho_G \\
&= \mathcal{B}_\tau[\![y_1{}^\smile b_1]\!]e\rho_G
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{unless-false}\ y_1\ y_2\rangle :: b_1]\!]e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{unless-false}\ y_1\ y_2\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{unless-false}\ y_1\ y_2\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\pi.\mathit{if_truish}(\mathcal{Y}_\tau[\![y_1]\!]e\rho\pi)(\mathcal{Y}_\tau[\![y_2]\!]e\rho\pi))e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)
\end{aligned}$$

$$\begin{array}{rcl}
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathit{if_truish}(\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G))(\mathcal{Y}_\tau[\![y_2]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda\pi_1\pi_2\epsilon\epsilon^*\rho_R\psi.\mathit{truish}\ \epsilon \to \pi_1\epsilon\epsilon^*\rho_R\psi, \pi_2\epsilon\epsilon^*\rho_R\psi) \\
& & (\mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G))(\mathcal{Y}_\tau[\![y_2]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)) \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Y}_\tau[\![y_1]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![y_1 \smile b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{D}[\![\Sigma']\!]
\end{array}$$

Case 9: $R =$ Open Branch/False.

Let $\Sigma = \langle t, \langle \texttt{unless-false}\ y_1\ y_2\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $v = \mathit{false}$.

Then $R(\Sigma) = \Sigma' = \langle t, y_2 \smile b_1, v, a, u, k, s\rangle$.

The proof of this case is to the proof of Case 8 as the proof of Case 7 is to the proof of Case 6.

Case 10: $R =$ Make Continuation.

Let $\Sigma = \langle t, \langle \texttt{make-cont}\ b_1\ \#a\rangle :: b_2, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_2, v, \langle\rangle, u, k', s\rangle$,
where $k' = \langle \textsc{cont}\ t\ b_1\ a\ u\ k\rangle$.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{make-cont}\ b_1\ \#a\rangle :: b_2]\!]e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathit{make_cont}(\mathcal{B}_\tau[\![b_1]\!]e\rho)\#a(\mathcal{B}_\tau[\![b_2]\!]e\rho))e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathit{make_cont}(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\#a(\mathcal{B}_\tau[\![b_2]\!]e\rho_G) \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda\pi'\nu\pi\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* = \nu \to \pi\epsilon\langle\rangle\rho_R(\lambda\epsilon.\pi'\epsilon\epsilon^*\rho_R\psi), \\
& & \mathit{wrong}\ \text{“bad stack”})(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\#a(\mathcal{B}_\tau[\![b_2]\!]e\rho_G)
\end{array}$$

$$
\begin{aligned}
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& (\#\mathcal{D}_a[\![a]\!] = \#a \rightarrow \mathcal{B}_\tau[\![b_2]\!]e\rho_G\mathcal{D}_v[\![v]\!]\langle\rangle\mathcal{D}_u[\![u]\!] \\
& (\lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]), \mathit{wrong} \text{ “bad stack”})\mathcal{D}_s[\![s]\!] \\
= \;& \mathcal{B}_\tau[\![b_2]\!]e\rho_G\mathcal{D}_v[\![v]\!]\langle\rangle\mathcal{D}_u[\![u]\!](\lambda\epsilon.(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]) \\
& \mathcal{D}_s[\![s]\!] \\
= \;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 11: $R =$ Literal.

Let $\Sigma = \langle t, \langle \texttt{literal}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $e(n) = \langle \texttt{constant}\ c\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, c, a, u, k, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] \;=\;& \mathcal{B}_\tau[\![\langle \texttt{literal}\ n\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{literal}\ n\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{Z}_\tau[\![\langle \texttt{literal}\ n\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho.\mathit{literal}(\mathcal{K}[\![(e(n))(1)]\!])e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathit{literal}(\mathcal{K}[\![(e(n))(1)]\!])(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda\epsilon'\pi\epsilon\epsilon^*\rho_R\psi.\pi\epsilon'\epsilon^*\rho_R\psi)\mathcal{K}[\![c]\!](\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{B}_\tau[\![b_1]\!]e\rho_G\mathcal{K}[\![c]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 12: $R =$ Closure.

Let $\Sigma = \langle t, \langle \texttt{closure}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $e(n) = \langle \texttt{template}\ b_2\ e_2\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v', a, u, k, s'\rangle$,
where $v' = \langle \text{CLOSURE}\ e(n)\ u\ \#s\rangle$ and $s' = s \frown \langle \text{NOT-SPECIFIED}\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{closure}\ n\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{closure}\ n\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{closure}\ n\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{closure}(\mathcal{T}_\tau[\![e(n)]\!]\rho)) e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{closure}(\mathcal{T}_\tau[\![e(n)]\!]\rho_G))(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\pi'\pi\epsilon\epsilon^*\rho_R\psi\sigma.\pi(\mathit{fix}(\lambda\epsilon.\langle \mathit{new}\,\sigma, \lambda\epsilon^*\kappa.\pi'\epsilon\epsilon^*\rho_R(\lambda\epsilon.\kappa\langle\epsilon\rangle)\rangle \text{ in } \mathtt{E})) \\
&\quad\ \epsilon^*\rho_R\psi(\mathit{update}(\mathit{new}\ \sigma)(\mathit{unspecified} \text{ in } \mathtt{E})\ \sigma)) \\
&\quad\ (\mathcal{B}_\tau[\![b_2]\!] e_2\rho_G)(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad\ (\mathit{fix}(\lambda\epsilon.\langle \mathit{new}\,\mathcal{D}_s[\![s]\!], \lambda\epsilon^*\kappa.\mathcal{B}_\tau[\![b_2]\!] e_2\rho_G\epsilon\epsilon^*\mathcal{D}_u[\![u]\!](\lambda\epsilon.\kappa\langle\epsilon\rangle)\rangle \text{ in } \mathtt{E})) \\
&\quad\ \mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!](\mathit{update}(\mathit{new}\ \mathcal{D}_s[\![s]\!])(\mathit{unspecified} \text{ in } \mathtt{E})\ \mathcal{D}_s[\![s]\!]) \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G\mathcal{D}_v[\![\langle \text{CLOSURE}\ e(n)\ u\ (\mathit{new}\,\mathcal{D}_s[\![s]\!])\rangle]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ (\mathit{update}(\mathit{new}\ \mathcal{D}_s[\![s]\!])(\mathit{unspecified} \text{ in } \mathtt{E})\ \mathcal{D}_s[\![s]\!]) \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G\mathcal{D}_v[\![v']\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ \mathcal{D}_s[\![s]\!][(\mathit{unspecified} \text{ in } \mathtt{E})/\#\mathcal{D}_s[\![s]\!]] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 13: $R =$ Global.

Let $\Sigma = \langle t, \langle \texttt{global}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$ and $e(n) = \langle \texttt{global-variable}\ i\rangle$, $\rho_G(i) = l$, and $v_1 = s(l) \neq$ UNDEFINED.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v_1, a, u, k, s\rangle$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{global}\ n\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{global}\ n\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{global}\ n\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{global}(\mathit{lookup}\ \rho\ (e(n))(1))) e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{global}(\mathit{lookup}\ \rho_G\ (e(n))(1))(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\pi\epsilon\epsilon^*\rho_R\psi.\mathit{hold}\ \alpha(\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \pi\epsilon\epsilon^*\rho_R\psi, \\
&\quad \mathit{wrong}\ \text{“undefined variable”}))) \\
&\quad (\rho_G i)(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{hold}\ l(\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad \epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \mathit{wrong}\ \text{“undefined variable”}))\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\kappa\sigma.\alpha < \#\sigma \to \mathit{send}(\sigma\alpha)\kappa\sigma, \mathit{empty}\ \text{in}\ \texttt{E}) \\
&\quad l(\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad \epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \mathit{wrong}\ \text{“undefined variable”}))\mathcal{D}_s[\![s]\!] \\
&= \mathit{send}(\mathcal{D}_s[\![s]\!] l)(\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad \epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \mathit{wrong}\ \text{“undefined variable”}))\mathcal{D}_s[\![s]\!] \\
&= (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad \epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \mathit{wrong}\ \text{“undefined variable”}))\langle\mathcal{D}_s[\![s]\!] l\rangle\mathcal{D}_s[\![s]\!] \\
&= (\mathcal{D}_s[\![s]\!] l \neq (\mathit{empty}\ \text{in}\ \texttt{E}) \to \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad (\mathcal{D}_s[\![s]\!] l)\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \mathit{wrong}\ \text{“undefined variable”})\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathcal{D}_s[\![s]\!] l)\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 14: $R =$ Set Global.

Let $\Sigma = \langle t, \langle \texttt{set-global!}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$, $e(n) = \langle \texttt{global-variable}\ i\rangle$, $\rho_G(i) = l$, and $l \leq \#s$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, \text{NOT-SPECIFIED}, a, u, k, s'\rangle$,
where $s' = s + \{l \mapsto v\}$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{set-global!}\ n\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{set-global!}\ n\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{set-global!}\ n\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{set\_global}(\mathit{lookup}\ \rho\ ((e(n))(1)))) e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{set\_global}(\mathit{lookup}\ \rho_G\ ((e(n))(1)))(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\pi\epsilon\epsilon^*\rho_R\psi.\mathit{assign}\ \alpha\epsilon(\pi\ (\mathit{unspecified}\ \text{in}\ \texttt{E})\ \epsilon^*\rho_R\psi)) \\
&\quad\ (\rho_G i)(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{assign}\ l\mathcal{D}_v[\![v]\!] \\
&\quad\ (\mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\epsilon\theta\sigma.\theta(\mathit{update}\ \alpha\epsilon\sigma)) \\
&\quad\ l\mathcal{D}_v[\![v]\!](\mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ (\mathit{update}\ l\mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!]) \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ \mathcal{D}_s[\![s]\!][\mathcal{D}_v[\![v]\!]/l] \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s']\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 15: $R = $ Local.

Let $\Sigma = \langle t, \langle \texttt{local}\ n_1\ n_2 \rangle :: b_1, v, a, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$, $\mathit{env\text{-}reference}(u, n_1, n_2) = l$, and $v_1 = s(l) \neq$ UNDEFINED.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v_1, a, u, k, s \rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{local}\ n_1\ n_2 \rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho. \mathcal{Z}_\tau[\![\langle \texttt{local}\ n_1\ n_2 \rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{local}\ n_1\ n_2 \rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho. \mathit{local}\, n_1 n_2) e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{local}\, n_1 n_2(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda \nu_1\nu_2\pi\epsilon\epsilon^*\rho_R\psi. \mathit{hold}(\rho_R\nu_1\nu_2)(\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \\
&\quad\ \pi\epsilon\epsilon^*\rho_R\psi, \mathit{wrong} \text{ “undefined variable”}))) \\
&\quad\ n_1 n_2(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{hold}(\mathcal{D}_u[\![u]\!] n_1 n_2) \\
&\quad\ (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”}))\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\kappa\sigma.\alpha < \#\sigma \rightarrow \mathit{send}(\sigma\alpha)\kappa\sigma, \mathit{empty} \text{ in } \texttt{E}) l \\
&\quad\ (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”}))\mathcal{D}_s[\![s]\!] \\
&= \mathit{send}(\mathcal{D}_s[\![s]\!] l) \\
&\quad\ (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”})\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon\kappa.\kappa\langle\epsilon\rangle)(\mathcal{D}_s[\![s]\!] l) \\
&\quad\ (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”})\mathcal{D}_s[\![s]\!]) \\
&= (\mathit{single}(\lambda\epsilon.\epsilon \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G\epsilon\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”})\langle\mathcal{D}_s[\![s]\!] l\rangle\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}_s[\![s]\!] l \neq (\mathit{empty} \text{ in } \texttt{E}) \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\mathcal{D}_s[\![s]\!] l)\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
&\quad\ \mathit{wrong} \text{ “undefined variable”})\mathcal{D}_s[\![s]\!]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathcal{D}_s[\![s]\!]l)\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 16: $R =$ Set Local.

Let $\Sigma = \langle t, \langle \texttt{set-local!}\ n_1\ n_2\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$, $\mathit{env\text{-}reference}(u, n_1, n_2) = l$, and $l \leq \#s$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, \text{NOT-SPECIFIED}, a, u, k, s'\rangle$,
where $s' = s + \{l \mapsto v\}$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{set-local!}\ n_1\ n_2\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{set-local!}\ n_1\ n_2\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{set-local!}\ n_1\ n_2\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{set\_local}\ n_1 n_2)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{set\_local}\ n_1 n_2(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda \nu_1\nu_2\pi\epsilon\epsilon^*\rho_R\psi.\mathit{assign}(\rho_R\nu_1\nu_2)\epsilon(\pi(\mathit{unspecified}\ \text{in}\ \texttt{E})\epsilon^*\rho_R\psi)) \\
&\quad\ n_1 n_2(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{assign}(\mathcal{D}_u[\![u]\!]n_1 n_2)\mathcal{D}_v[\![v]\!](\mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E}) \\
&\quad\ \mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\epsilon\theta\sigma.\theta(\mathit{update}\ \alpha\epsilon\sigma))l\mathcal{D}_v[\![v]\!](\mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E}) \\
&\quad\ \mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ (\mathit{update}\ l\ \mathcal{D}_v[\![v]\!]\mathcal{D}_s[\![s]\!]) \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!] \\
&\quad\ \mathcal{D}_s[\![s]\!][\mathcal{D}_v[\![v]\!]/l] \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \text{in}\ \texttt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s']\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 17: $R =$ Push.

Let $\Sigma = \langle t, \langle \texttt{push}\rangle :: b_1, v, a, u, k, s\rangle$, where $t = \langle \texttt{template}\ b\ e\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v, v :: a, u, k, s\rangle$.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{push}\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{push}\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{push}\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathit{push}) e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathit{push}(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda\pi\epsilon\epsilon^*\rho_R\psi.\pi\epsilon(\epsilon^*\frown\langle\epsilon\rangle)\rho_R\psi) \\
& & (\mathcal{B}_\tau[\![b_1]\!] e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!](\mathcal{D}_a[\![a]\!]\frown\langle\mathcal{D}_v[\![v]\!]\rangle)\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![v :: a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{D}[\![\Sigma']\!]
\end{array}$$

Case 18: $R =$ Alternate Make Environment.

Let $\Sigma = \langle t, \langle \texttt{make-env}\ \#a\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v, \langle\rangle, u', k, s\frown(\mathit{rev}\ a)\rangle$,
where $u' = \mathit{add\text{-}layer}'(u, \#s, \#a)$.

If $\mathit{tievals\_aux} = \lambda\nu_1\nu_2.\langle\nu_1\ \cdots\ (\nu_1+\nu_2-1)\rangle$, then

$$\mathit{tievals}\ \xi\epsilon^*\sigma = \xi(\mathit{tievals\_aux}(\#\sigma)(\#\epsilon^*))(\sigma\frown\epsilon^*).$$

Also,

$$\mathit{extend}_R\,\mathcal{D}_u[\![u]\!]\langle m\ \cdots\ (m+n-1)\rangle = \mathcal{D}_u[\![\mathit{add\text{-}layer}'(u, m, n)]\!].$$

These two results are used in the following derivation.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{make-env}\ \#a\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{make-env}\ \#a\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{make-env}\ \#a\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)
\end{array}$$

$$
\begin{array}{rl}
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= & (\lambda e\rho.\mathit{make\_env}\ \#a)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= & \mathit{make\_env}\ \#a(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= & (\lambda\nu\pi\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* = \nu \rightarrow \\
& \mathit{tievals}(\lambda\alpha^*.\pi\epsilon\langle\rangle(\mathit{extend}_R\,\rho_R\alpha^*)\psi)\epsilon^*, \mathit{wrong}\text{ “bad stack”}) \\
& \#a(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= & (\#\mathcal{D}_a[\![a]\!] = \#a \rightarrow \\
& \mathit{tievals}(\lambda\alpha^*.(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\langle\rangle(\mathit{extend}_R\,\mathcal{D}_u[\![u]\!]\alpha^*)\mathcal{D}_k[\![k]\!]) \\
& \mathcal{D}_a[\![a]\!], \mathit{wrong}\text{ “bad stack”})\mathcal{D}_s[\![s]\!] \\
= & \mathit{tievals}(\lambda\alpha^*.(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\langle\rangle(\mathit{extend}_R\,\mathcal{D}_u[\![u]\!]\alpha^*)\mathcal{D}_k[\![k]\!]) \\
& \mathcal{D}_a[\![a]\!]\mathcal{D}_s[\![s]\!] \\
= & (\lambda\alpha^*.(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\langle\rangle(\mathit{extend}_R\,\mathcal{D}_u[\![u]\!]\alpha^*)\mathcal{D}_k[\![k]\!]) \\
& \mathit{tievals\_aux}(\#\mathcal{D}_s[\![s]\!])(\#\mathcal{D}_a[\![a]\!])(\mathcal{D}_s[\![s]\!]^\frown\mathcal{D}_a[\![a]\!]) \\
= & \mathcal{B}_\tau[\![b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\langle\rangle \\
& (\mathit{extend}_R\,\mathcal{D}_u[\![u]\!]\langle\#\mathcal{D}_s[\![s]\!]\ \cdots\ (\#\mathcal{D}_s[\![s]\!] + \#\mathcal{D}_a[\![a]\!] - 1)\rangle) \\
& \mathcal{D}_k[\![k]\!](\mathcal{D}_s[\![s]\!]^\frown\mathcal{D}_a[\![a]\!]) \\
= & \mathcal{B}_\tau[\![b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\langle\rangle \\
& \mathcal{D}_u[\![\mathit{add\text{-}layer}'(u, \#\mathcal{D}_s[\![s]\!], \#\mathcal{D}_a[\![a]\!])]\!]\mathcal{D}_k[\![k]\!](\mathcal{D}_s[\![s]\!]^\frown\mathcal{D}_a[\![a]\!]) \\
= & \mathcal{D}[\![\Sigma']\!]
\end{array}
$$

Case 19: $R =$ Make Rest List.

Let $\Sigma = \langle t, \langle\texttt{make-rest-list}\ n_1\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle\texttt{template}\ b\ e\rangle$ and $\#a = n_1 + n_2$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v', a\dagger n_2, u, k, s'\rangle$,
where $v' = \mathit{mrl\text{-}value}(n_2, \mathit{null}, a, s)$ and $s' = \mathit{mrl\text{-}store}(n_2, \mathit{null}, a, s)$.

In the following derivation, we will use the fact that, if $\#a \geq n$, then $\mathit{list}\,\mathcal{D}_a[\![a\ddagger n]\!]\kappa\mathcal{D}_s[\![s]\!] =$

$$\kappa\langle\mathcal{D}_v[\![\mathit{mrl\text{-}value}(n, \mathit{null}, a, s)]\!]\rangle\mathcal{D}_s[\![\mathit{mrl\text{-}store}(n, \mathit{null}, a, s)]\!].$$

$$\mathcal{D}[\![\Sigma]\!] \quad = \quad \mathcal{B}_\tau[\![\langle\texttt{make-rest-list}\ n_1\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]$$

$$
\begin{aligned}
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{make-rest-list}\ n_1\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{make-rest-list}\ n_1\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathit{make\_rest\_list}\, n_1)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{make\_rest\_list}\, n_1(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\nu\pi\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* \geq \nu \rightarrow \\
&\quad \mathit{list}(\epsilon^*\dagger\nu)(\mathit{single}\ \lambda\epsilon.\pi\epsilon(\epsilon^*\ddagger\nu)\rho_R\psi), \\
&\quad \mathit{wrong}\ \text{“bad stack”}) \\
&\quad n_1(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\#\mathcal{D}_a[\![a]\!] \geq n_1 \rightarrow \\
&\quad \mathit{list}(\mathcal{D}_a[\![a]\!]\dagger n_1)(\mathit{single}\ \lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e\rho_G\epsilon(\mathcal{D}_a[\![a]\!]\ddagger n_1)\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]), \\
&\quad \mathit{wrong}\ \text{“bad stack”})\mathcal{D}_s[\![s]\!] \\
&= \mathit{list}(\mathcal{D}_a[\![a]\!]\dagger n_1)(\mathit{single}\ \lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e\rho_G\epsilon(\mathcal{D}_a[\![a]\!]\ddagger n_1)\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]) \\
&\quad \mathcal{D}_s[\![s]\!] \\
&= \mathit{list}\, \mathcal{D}_a[\![a\ddagger n_2]\!](\mathit{single}\ \lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e\rho_G\epsilon\mathcal{D}_a[\![a\dagger n_2]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= (\mathit{single}\ \lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e\rho_G\epsilon\mathcal{D}_a[\![a\dagger n_2]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]) \\
&\quad \langle\mathcal{D}_v[\![\mathit{mrl\text{-}value}(n_2, \mathit{null}, a, s)]\!]\rangle\mathcal{D}_s[\![\mathit{mrl\text{-}store}(n_2, \mathit{null}, a, s)]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G\mathcal{D}_v[\![\mathit{mrl\text{-}value}(n_2, \mathit{null}, a, s)]\!] \\
&\quad \mathcal{D}_a[\![a\dagger n_2]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![\mathit{mrl\text{-}store}(n_2, \mathit{null}, a, s)]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 20: $R =$ Unspecified.

Let $\Sigma = \langle\langle t, \langle\texttt{unspecified}\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \texttt{template}\ b\ e\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, \textsc{not-specified}, a, u, k, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\texttt{unspecified}\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle\texttt{unspecified}\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle\texttt{unspecified}\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)
\end{aligned}
$$

$$
\begin{aligned}
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& (\lambda e\rho.\mathit{literal}(\mathit{unspecified}\text{ in }\mathtt{E}))e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& \mathit{literal}\,(\mathit{unspecified}\text{ in }\mathtt{E})(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& (\lambda\epsilon'\pi\epsilon\epsilon^*\rho_R\psi.\pi\epsilon'\epsilon^*\rho_R\psi)(\mathit{unspecified}\text{ in }\mathtt{E})(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& (\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathit{unspecified}\text{ in }\mathtt{E})\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
= \;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 21: $R =$ Check Args $=$.

Let $\Sigma = \langle t, \langle\mathtt{checkargs=}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $t = \langle\mathtt{template}\ b\ e\rangle$ and $\#a = n$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v, a, u, k, s\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] \;=\;& \mathcal{B}_\tau[\![\langle\mathtt{checkargs=}\ n\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho.\mathcal{Z}_\tau[\![\langle\mathtt{checkargs=}\ n\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{Z}_\tau[\![\langle\mathtt{checkargs=}\ n\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho.\mathit{check\_args\_eq}\,n)e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathit{check\_args\_eq}\,n(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda\nu\pi\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* = \nu \rightarrow \pi\epsilon\epsilon^*\rho_R\psi, \\
& \mathit{wrong}\text{ “bad arg count”})n(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\#\mathcal{D}_a[\![a]\!] = n \rightarrow \mathcal{B}_\tau[\![b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!], \\
& \mathit{wrong}\text{ “bad arg count”})\mathcal{D}_s[\![s]\!] \\
=\;& (\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 22: $R =$ Check Args >=.

Let $\Sigma = \langle t, \langle \texttt{checkargs>=}\ n\rangle :: b_1, v, a, u, k, s\rangle$,
where $\#a \geq n$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v, a, u, k, s\rangle$.

The proof of this case is essentially the same as the proof for Case 21.

Case 23: $R =$ Primitive CWCC.

Let $\Sigma = \langle\langle \texttt{template}\ b\ e\rangle, \langle \texttt{\%\%cwcc}\rangle :: b_1, v, a, u, k, s\rangle$,
where $a = \langle v_1\rangle$ and $v_1 = \langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, t_1(1), v_1, a_1, u_1, k, s_1\rangle$,
where $a_1 = \langle\langle \text{ESCAPE}\ k\ \#s\rangle\rangle$ and $s_1 = s^\frown\langle \text{NOT-SPECIFIED}\rangle$.

$$
\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{\%\%cwcc}\rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{\%\%cwcc}\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{\%\%cwcc}\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi\sigma.\#\epsilon^* = 1 \rightarrow \\
& & \mathit{call}\ 1(\epsilon^*0)\langle \mathit{make\_escape}\ \psi\sigma\rangle\rho_R\psi \\
& & (\mathit{update}\ (\mathit{new}\ \sigma)(\mathit{unspecified}\ \text{in}\ \mathtt{E})\sigma), \\
& & \mathit{wrong}\ \text{“bad arg count”}\sigma) \\
& & e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \#\mathcal{D}_a[\![\langle\langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle\rangle]\!] = 1 \rightarrow \\
& & \mathit{call}\ 1(\mathcal{D}_a[\![\langle\langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle\rangle]\!]0) \\
& & \langle \mathit{make\_escape}\ \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle\mathcal{D}_k[\![k]\!]\mathcal{D}_u[\![u]\!] \\
& & (\mathit{update}\ (\mathit{new}\ \mathcal{D}_s[\![s]\!])(\mathit{unspecified}\ \text{in}\ \mathtt{E})\mathcal{D}_s[\![s]\!]), \\
& & \mathit{wrong}\ \text{“bad arg count”}\mathcal{D}_s[\![s]\!] \\
& = & \mathit{call}\ 1\mathcal{D}_v[\![\langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle]\!]\langle \mathit{make\_escape}\ \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \\
& & \mathcal{D}_k[\![k]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_s[\![s_1]\!] \\
& = & \mathit{applicate}\ \mathcal{D}_v[\![\langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle]\!] \\
& & \langle \mathit{make\_escape}\ \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle(\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
& = & (\lambda\epsilon^*\kappa.\mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![\langle \text{CLOSURE}\ t_1\ u_1\ l_1\rangle]\!]\epsilon^*\mathcal{D}_u[\![u_1]\!](\lambda\epsilon.\kappa\langle\epsilon\rangle))
\end{array}
$$

$$
\begin{aligned}
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle(\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v_1]\!] \\
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \\
& \mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v_1]\!] \\
& \langle(\lambda\psi\sigma.\langle(\mathit{new}\,\sigma), \mathit{single\_arg}(\lambda\epsilon\kappa.\psi\epsilon)\rangle \text{ in } \mathtt{E})\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \\
& \mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v_1]\!] \\
& \langle\langle \#\mathcal{D}_s[\![s]\!], \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![k]\!]\epsilon)\rangle \text{ in } \mathtt{E}\rangle \\
& \mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v_1]\!]\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 24: $R =$ Primitive CWCC-Escape.

Let $\Sigma = \langle\langle \mathtt{template}\ b\ e\rangle, \langle \mathtt{\%\%cwcc}\rangle :: b_0, v, a, u, k, s\rangle$, where $a = \langle\langle \textsc{escape}\ \langle \textsc{cont}\ t_1\ b_1\ a_1\ u_1\ k_1\rangle\ l_1\rangle\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, b_1, v_1, a_1, u_1, k_1, s_1\rangle$, where $t_1 = \langle \mathtt{template}\ b'\ e_1\rangle$, $v_1 = \langle \textsc{escape}\ k\ \#s\rangle$, and $s_1 = s ^\frown \langle \textsc{not-specified}\rangle$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] \;=\;& \mathcal{B}_\tau[\![\langle \mathtt{\%\%cwcc}\rangle :: b_0]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \mathtt{\%\%cwcc}\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_0]\!]e\rho))e\rho_G \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \mathcal{Z}_\tau[\![\langle \mathtt{\%\%cwcc}\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_0]\!]e\rho_G) \\
& \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi\sigma.\#\epsilon^* = 1 \rightarrow \\
& \mathit{call}\, 1(\epsilon^*0)\langle \mathit{make\_escape}\, \psi\sigma\rangle\rho_R\psi \\
& (\mathit{update}\,(\mathit{new}\,\sigma)(\mathit{unspecified} \text{ in } \mathtt{E})\sigma), \\
& \mathit{wrong}\text{ “bad arg count”}\sigma) \\
& e\rho_G(\mathcal{B}_\tau[\![b_0]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
=\;& \#\mathcal{D}_a[\![\langle\langle \textsc{escape}\ \langle \textsc{cont}\ t_1\ b_1\ a_1\ u_1\ k_1\rangle\ l_1\rangle\rangle]\!] = 1 \rightarrow
\end{aligned}
$$

$$
\begin{aligned}
& \mathit{call}\, 1(\mathcal{D}_a[\![\langle\langle \text{ESCAPE } \langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle \ l_1\rangle\rangle]\!]0) \\
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \mathcal{D}_k[\![k]\!]\mathcal{D}_u[\![u]\!] \\
& (\mathit{update}\, (\mathit{new}\, \mathcal{D}_s[\![s]\!])(\mathit{unspecified} \text{ in } \mathtt{E})\mathcal{D}_s[\![s]\!]), \\
& \mathit{wrong} \text{ “bad arg count”} \mathcal{D}_s[\![s]\!] \\
= \;& \mathit{call}\, 1\mathcal{D}_v[\![\langle \text{ESCAPE } \langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle \ l_1\rangle]\!] \\
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \mathcal{D}_k[\![k]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathit{applicate}\, \mathcal{D}_v[\![\langle \text{ESCAPE } \langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle \ l_1\rangle]\!] \\
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle (\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\epsilon) \\
& \langle \mathit{make\_escape}\, \mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \\
& (\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\epsilon) \\
& \langle (\lambda\psi\sigma.\langle (\mathit{new}\, \sigma), \mathit{single\_arg}(\lambda\epsilon\kappa.\psi\epsilon)\rangle \text{ in } \mathtt{E})\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]\rangle \\
& (\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\epsilon) \\
& \langle\langle \#\mathcal{D}_s[\![s]\!], \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![k]\!]\epsilon)\rangle \text{ in } \mathtt{E}\rangle \\
& (\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathit{single\_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\epsilon) \\
& \mathcal{D}_a[\![\langle v_1\rangle]\!](\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& (\lambda\epsilon\kappa.\mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\epsilon)\mathcal{D}_v[\![v_1]\!](\mathit{single}\, \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{D}_k[\![\langle \text{CONT } t_1 \ b_1 \ a_1 \ u_1 \ k_1\rangle]\!]\mathcal{D}_v[\![v_1]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& (\lambda\epsilon.\mathcal{B}_\tau[\![b_1]\!]e_1\rho_G\epsilon\mathcal{D}_a[\![a_1]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k_1]\!])\mathcal{D}_v[\![v_1]\!]\mathcal{D}_s[\![s_1]\!] \\
= \;& \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 25: $R =$ Primitive CWCC-Escape-Halt.

Let $\Sigma = \langle t, \langle \mathtt{\%\%cwcc}\rangle :: b_1, v, a, u, k, s\rangle$,
where $a = \langle\langle \text{ESCAPE HALT } l\rangle\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, \langle\rangle, v', a, u, k, s\rangle$,
where $v' = \langle \text{ESCAPE } k \ \#s\rangle$.

$\mathcal{O}'[\![\Sigma']\!]$ is undefined since $v' \notin \mathtt{R}$. Thus $\mathcal{O}'[\![\Sigma]\!]$ is undefined, and so we do not have to consider this case.

Case 26: $R =$ Primitive Cons.

Let $\Sigma = \langle t, \langle \texttt{\%\%cons}\rangle :: b_1, v, \langle v_1\ v_2\rangle ^\frown a_1, u, k, s\rangle$,
where $t = \langle \texttt{template}\ b\ e\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v', \langle\rangle, u, k, s ^\frown \langle v_2\ v_1\rangle\rangle$,
where $v' = \langle \text{MUTABLE-PAIR}\ \#s\ (1 + \#s)\rangle$.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{\%\%cons}\rangle :: b_1]\!] e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{\%\%cons}\rangle]\!] e\rho(\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{\%\%cons}\rangle]\!] e\rho_G(\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
& & \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi\sigma.\#\epsilon^* \geq 2 \rightarrow \\
& & (\lambda\sigma'.\pi(\langle \mathit{new}\ \sigma, \mathit{new}\ \sigma', \mathit{mutable}\rangle \text{ in } \texttt{E}) \\
& & \langle\rangle\rho_R\psi(\mathit{update}(\mathit{new}\ \sigma')((\mathit{rev}\ \epsilon^*)0)\sigma')) \\
& & (\mathit{update}(\mathit{new}\ \sigma)((\mathit{rev}\ \epsilon^*)1)\sigma), \mathit{wrong}\ \text{“bad arg count”}\sigma) \\
& & e\rho_G\mathcal{B}_\tau[\![b_1]\!] e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
& = & \#\mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!] \geq 2 \rightarrow \\
& & (\lambda\sigma'.(\mathcal{B}_\tau[\![b_1]\!] e\rho_G)(\langle \mathit{new}\ \mathcal{D}_s[\![s]\!], \mathit{new}\ \sigma', \mathit{mutable}\rangle \text{ in } \texttt{E}) \\
& & \langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!](\mathit{update}(\mathit{new}\ \sigma')((\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!])0)\sigma')) \\
& & (\mathit{update}(\mathit{new}\ \mathcal{D}_s[\![s]\!])(\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ v_2\rangle ^\frown a_1]\!]1), \\
& & \mathit{wrong}\ \text{“bad arg count”}\mathcal{D}_s[\![s]\!] \\
& = & (\lambda\sigma'.\mathcal{B}_\tau[\![b_1]\!] e\rho_G(\langle \#\mathcal{D}_s[\![s]\!], \mathit{new}\ \sigma', \mathit{mutable}\rangle \text{ in } \texttt{E}) \\
& & \langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!](\mathit{update}(\mathit{new}\ \sigma')D_v[\![v_1]\!]\sigma'))\mathcal{D}_s[\![s]\!] ^\frown \langle\mathcal{D}_v[\![v_2]\!]\rangle \\
& = & \mathcal{B}_\tau[\![b_1]\!] e\rho_G(\langle \#\mathcal{D}_s[\![s]\!], (1 + \#\mathcal{D}_s[\![s]\!]), \mathit{mutable}\rangle \text{ in } \texttt{E}) \\
& & \langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] ^\frown \langle D_v[\![v_2]\!]\ D_v[\![v_1]\!]\rangle \\
& = & \mathcal{D}[\![\Sigma']\!]
\end{array}$$

Case 27: $R =$ Primitive Car-Immutable Pair.

Let $\Sigma = \langle t, \langle \texttt{\%\%car} \rangle :: b_1, v, a, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$ and $a = \langle \texttt{immutable-pair}\ c_1\ c_2 \rangle :: a_1$

Then $R(\Sigma) = \Sigma' = \langle t, b_1, c_1, \langle\rangle, u, k, s \rangle$.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{\%\%car} \rangle :: b_1]\!] e \rho_G \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho . \mathcal{Z}_\tau[\![\langle \texttt{\%\%car} \rangle]\!] e\rho (\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{\%\%car} \rangle]\!] e\rho_G (\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & (\lambda e\rho\pi\epsilon\epsilon^* \rho_R \psi\sigma . \# \epsilon^* \geq 1 \rightarrow \\
& & (\mathit{rev}\ \epsilon^*)0 : \texttt{E}_\mathrm{p} \rightarrow \pi(\sigma(((\mathit{rev}\ \epsilon^*)0 | \texttt{E}_\mathrm{p})0)) \langle\rangle \rho_R \psi\sigma, \\
& & \mathit{wrong}\ \text{“attempt to take car of non-pair”} \sigma, \\
& & \mathit{wrong}\ \text{“bad arg count”} \sigma) \\
& & e\rho_G (\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \# \mathcal{D}_a[\![\langle \texttt{immutable-pair}\ c_1\ c_2 \rangle :: a_1]\!] \geq 1 \rightarrow \\
& & (\mathit{rev}\ \mathcal{D}_a[\![\langle \texttt{immutable-pair}\ c_1\ c_2 \rangle :: a_1]\!])0 : \texttt{E}_\mathrm{p} \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
& & (\mathcal{D}_s[\![s]\!](((\mathit{rev}\ \mathcal{D}_a[\![\langle \texttt{immutable-pair}\ c_1\ c_2 \rangle :: a_1]\!])0 | \texttt{E}_\mathrm{p})0)) \\
& & \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!], \\
& & \mathit{wrong}\ \text{“attempt to take car of non-pair”} \mathcal{D}_s[\![s]\!], \\
& & \mathit{wrong}\ \text{“bad arg count”} \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e\rho_G (\mathcal{D}_s[\![s]\!](\mathcal{D}_v[\![\langle \texttt{immutable-pair}\ c_1\ c_2 \rangle]\!]0)) \\
& & \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e\rho_G c_1 \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{D}[\![\Sigma']\!]
\end{array}$$

Case 28: $R =$ Primitive Car-Mutable Pair.

Let $\Sigma = \langle t, \langle \texttt{\%\%car} \rangle :: b_1, v, a, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$ and $a = \langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle :: a_1$

Then $R(\Sigma) = \Sigma' = \langle t, b_1, s(l_1), \langle\rangle, u, k, s \rangle$.

$$
\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{\%\%car} \rangle :: b_1]\!] e \rho_G \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & (\lambda e \rho . \mathcal{Z}_\tau[\![\langle \texttt{\%\%car} \rangle]\!] e \rho (\mathcal{B}_\tau[\![b_1]\!] e \rho)) e \rho_G \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{Z}_\tau[\![\langle \texttt{\%\%car} \rangle]\!] e \rho_G (\mathcal{B}_\tau[\![b_1]\!] e \rho_G) \\
& & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & (\lambda e \rho \pi \epsilon \epsilon^* \rho_R \psi \sigma . \# \epsilon^* \geq 1 \rightarrow \\
& & (\mathit{rev}\ \epsilon^*) 0 : \texttt{E}_\text{p} \rightarrow \pi (\sigma (((\mathit{rev}\ \epsilon^*) 0 | \texttt{E}_\text{p}) 0)) \langle\rangle \rho_R \psi \sigma, \\
& & \mathit{wrong}\ \text{“attempt to take car of non-pair”} \sigma, \\
& & \mathit{wrong}\ \text{“bad arg count”} \sigma) \\
& & e \rho_G (\mathcal{B}_\tau[\![b_1]\!] e \rho_G) \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \# \mathcal{D}_a[\![\langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle :: a_1]\!] \geq 1 \rightarrow \\
& & (\mathit{rev}\ \mathcal{D}_a[\![\langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle :: a_1]\!]) 0 : \texttt{E}_\text{p} \rightarrow \mathcal{B}_\tau[\![b_1]\!] e \rho_G \\
& & (\mathcal{D}_s[\![s]\!] (((\mathit{rev}\ \mathcal{D}_a[\![\langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle :: a_1]\!]) 0 | \texttt{E}_\text{p}) 0)) \\
& & \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!], \\
& & \mathit{wrong}\ \text{“attempt to take car of non-pair”} \mathcal{D}_s[\![s]\!], \\
& & \mathit{wrong}\ \text{“bad arg count”} \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e \rho_G (\mathcal{D}_s[\![s]\!] (\mathcal{D}_v[\![\langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle]\!] 0)) \\
& & \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{B}_\tau[\![b_1]\!] e \rho_G (\mathcal{D}_s[\![s]\!] l_1) \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
& = & \mathcal{D}[\![\Sigma']\!]
\end{array}
$$

Case 29: $R =$ Primitive Set-Car!.

Let $\Sigma = \langle t, \langle \texttt{\%\%set-car!} \rangle :: b_1, v, a, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$, $a = \langle v_1\ \langle \text{MUTABLE-PAIR}\ l_1\ l_2 \rangle \rangle \frown a_1$, and $l_1 < \# s$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, \text{NOT-SPECIFIED}, \langle\rangle, u, k, s' \rangle$,
where $s' = s + \{ l_1 \mapsto v_1 \}$.

$$
\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\texttt{\%\%set-car!}\rangle :: b_1]\!]e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle\texttt{\%\%set-car!}\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle\texttt{\%\%set-car!}\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi.\#\epsilon^* \geq 2 \to (\mathit{rev}\ \epsilon^*)1 : \mathtt{E}_\mathrm{p} \to \\
&\quad ((\mathit{rev}\ \epsilon^*)1|\mathtt{E}_\mathrm{p})2 = \mathit{mutable} \to \\
&\quad \mathit{assign}(((\mathit{rev}\ \epsilon^*)1|\mathtt{E}_\mathrm{p})0)((\mathit{rev}\ \epsilon^*)0)(\pi(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\rho_R\psi), \\
&\quad \mathit{wrong}\ \text{“attempt to set car of immutable pair”}, \\
&\quad \mathit{wrong}\ \text{“attempt to set car of non-pair”}, \\
&\quad \mathit{wrong}\ \text{“bad arg count”}) \\
&\quad e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\#\mathcal{D}_a[\![\langle v_1\ \langle\textsc{mutable-pair}\ l_1\ l_2\rangle\rangle ^\frown a_1]\!] \geq 2 \to \\
&\quad (\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ \langle\textsc{mutable-pair}\ l_1\ l_2\rangle\rangle ^\frown a_1]\!])1 : \mathtt{E}_\mathrm{p} \to \\
&\quad ((\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ \langle\textsc{mutable-pair}\ l_1\ l_2\rangle\rangle ^\frown a_1]\!])1|\mathtt{E}_\mathrm{p})2 = \mathit{mutable} \to \\
&\quad \mathit{assign}(((\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ \langle\textsc{mutable-pair}\ l_1\ l_2\rangle\rangle ^\frown a_1]\!])1|\mathtt{E}_\mathrm{p})0) \\
&\quad ((\mathit{rev}\ \mathcal{D}_a[\![\langle v_1\ \langle\textsc{mutable-pair}\ l_1\ l_2\rangle\rangle ^\frown a_1]\!])0) \\
&\quad ((\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]), \\
&\quad \mathit{wrong}\ \text{“attempt to set car of immutable pair”}, \\
&\quad \mathit{wrong}\ \text{“attempt to set car of non-pair”}, \\
&\quad \mathit{wrong}\ \text{“bad arg count”})\mathcal{D}_s[\![s]\!] \\
&= \mathit{assign}\ l_1\mathcal{D}_v[\![v_1]\!]((\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]) \\
&\quad \mathcal{D}_s[\![s]\!] \\
&= (\lambda\alpha\epsilon\theta\sigma.\theta(\mathit{update}\ \alpha\epsilon\sigma)) \\
&\quad l_1\mathcal{D}_v[\![v_1]\!]((\mathcal{B}_\tau[\![b_1]\!]e\rho_G)(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!](\mathit{update}\ l_1\mathcal{D}_v[\![v_1]\!]\mathcal{D}_s[\![s]\!]) \\
&= \mathcal{B}_\tau[\![b_1]\!]e\rho_G(\mathit{unspecified}\ \mathrm{in}\ \mathtt{E})\langle\rangle\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s']\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}
$$

Case 30: $R =$ Primitive Apply-Closure.

Let $\Sigma = \langle\langle\texttt{template}\ b\ e\rangle, \langle\texttt{\%\%apply}\rangle :: b_1, v, a, u, k, s\rangle$,
where $a = \langle v_1\rangle^\frown a_1{}^\frown\langle v_2\rangle$, $v_2 = \langle\textsc{closure}\ t_1\ u_1\ l_1\rangle$, and $t_1 = \langle\texttt{template}\ b_2\ e_2\rangle$.

Then $R(\Sigma) = \Sigma' = \langle t_1, b_2, v_2, a_2, u_1, k, s\rangle$,
where $a_2 = \mathit{app\text{-}stack}(v_1, a_1, s)$.

In the following derivation, we use the fact that, if $\mathit{app\text{-}stack}(v, a, s)$ is defined, then

$$\mathit{app\text{-}stack}(v, a, s) = \mathit{list\_to\_seq}\,\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_s[\![s]\!].$$

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle\texttt{\%\%apply}\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle\texttt{\%\%apply}\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle\texttt{\%\%apply}\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi\sigma.\#\epsilon^* \geq 2 \rightarrow \\
&\quad\ (\lambda\epsilon^*\epsilon.\mathit{call}(\#\epsilon^*)\epsilon\epsilon^*\rho_R\psi\sigma) \\
&\quad\ (\mathit{list_to_seq}((\mathit{rev}\ \epsilon^*)0)(((\mathit{rev}\ \epsilon^*)\dagger 1)\ddagger(\#\epsilon^* - 2))\sigma) \\
&\quad\ ((\mathit{rev}\ \epsilon^*)(\#\epsilon^* - 1)), \mathit{wrong}\ \text{“bad arg count”}\sigma) \\
&\quad\ e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon^*\epsilon.\mathit{call}(\#\epsilon^*)\epsilon\epsilon^*\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]) \\
&\quad\ (\mathit{list_to_seq}\,\mathcal{D}_v[\![v_1]\!]\mathcal{D}_a[\![a_1]\!]\mathcal{D}_s[\![s]\!])\mathcal{D}_v[\![v_2]\!] \\
&= (\lambda\epsilon^*\epsilon.\mathit{call}(\#\epsilon^*)\epsilon\epsilon^*\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!])\mathcal{D}_a[\![a_2]\!]\mathcal{D}_v[\![v_2]\!] \\
&= \mathit{call}(\#\mathcal{D}_a[\![a_2]\!])\mathcal{D}_v[\![v_2]\!]\mathcal{D}_a[\![a_2]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathit{applicate}\,\mathcal{D}_v[\![v_2]\!]\mathcal{D}_a[\![a_2]\!](\mathit{single}\,\mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon^*\kappa.\mathcal{T}_\tau[\![t_1]\!]\rho_G\mathcal{D}_v[\![v_2]\!]\epsilon^*\mathcal{D}_u[\![u_1]\!](\lambda\epsilon.\kappa\langle\epsilon\rangle)) \\
&\quad\ \mathcal{D}_a[\![a_2]\!](\mathit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_2]\!]e_2\rho_G\mathcal{D}_v[\![v_2]\!]\mathcal{D}_a[\![a_2]\!]\mathcal{D}_u[\![u_1]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 31: $R =$ Primitive Apply-Escape.

Let $\Sigma = \langle\langle \texttt{template}\ b\ e\rangle, \langle \texttt{\%\%apply}\rangle :: b_1, v, a, u, k, s\rangle$,
where $a = \langle v_1\rangle \frown a_1 \frown \langle v_2\rangle$, $v_2 = \langle \text{ESCAPE}\ k_1\ l_1\rangle$, $k_1 = \langle \text{CONT}\ t_2\ b_2\ a_2\ u_2\ k_2\rangle$,
$t_2 = \langle \texttt{template}\ b_3\ e_2\rangle$, and $\langle v_3\rangle = \textit{app-stack}(v_1, a_1, s)$.

Then $R(\Sigma) = \Sigma' = \langle t_2, b_2, v_3, a_2, u_2, k_2, s\rangle$.

In the following derivation, we use the fact that, if $\textit{app-stack}(v, a, s)$ is defined, then

$$\textit{app-stack}(v, a, s) = \textit{list\_to\_seq}\, \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_s[\![s]\!].$$

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{\%\%apply}\rangle :: b_1]\!]e\rho_G\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho.\mathcal{Z}_\tau[\![\langle \texttt{\%\%apply}\rangle]\!]e\rho(\mathcal{B}_\tau[\![b_1]\!]e\rho))e\rho_G \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{\%\%apply}\rangle]\!]e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G) \\
&\quad\ \mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi\sigma.\#\epsilon^* \geq 2 \rightarrow \\
&\quad\ (\lambda\epsilon^*\epsilon.\textit{call}(\#\epsilon^*)\epsilon\epsilon^*\rho_R\psi\sigma) \\
&\quad\ (\textit{list_to_seq}((\textit{rev}\ \epsilon^*)0)(((\textit{rev}\ \epsilon^*)\dagger 1)\ddagger(\#\epsilon^* - 2))\sigma) \\
&\quad\ ((\textit{rev}\ \epsilon^*)(\#\epsilon^* - 1)), \textit{wrong}\ \text{“bad arg count”}\sigma) \\
&\quad\ e\rho_G(\mathcal{B}_\tau[\![b_1]\!]e\rho_G)\mathcal{D}_v[\![v]\!]\mathcal{D}_a[\![a]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon^*\epsilon.\textit{call}(\#\epsilon^*)\epsilon\epsilon^*\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!]) \\
&\quad\ (\textit{list_to_seq}\,\mathcal{D}_v[\![v_1]\!]\mathcal{D}_a[\![a_1]\!]\mathcal{D}_s[\![s]\!])\mathcal{D}_v[\![v_2]\!] \\
&= (\lambda\epsilon^*\epsilon.\textit{call}(\#\epsilon^*)\epsilon\epsilon^*\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!])\mathcal{D}_a[\![\langle v_3\rangle]\!]\mathcal{D}_v[\![v_2]\!] \\
&= \textit{call}(\#\mathcal{D}_a[\![\langle v_3\rangle]\!])\mathcal{D}_v[\![v_2]\!]\mathcal{D}_a[\![\langle v_3\rangle]\!]\mathcal{D}_u[\![u]\!]\mathcal{D}_k[\![k]\!]\mathcal{D}_s[\![s]\!] \\
&= \textit{applicate}\ \mathcal{D}_v[\![v_2]\!]\mathcal{D}_a[\![\langle v_3\rangle]\!](\textit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \textit{single_arg}(\lambda\epsilon\kappa.\mathcal{D}_k[\![k_1]\!]\epsilon)\mathcal{D}_a[\![\langle v_3\rangle]\!](\textit{single}\ \mathcal{D}_k[\![k]\!])\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}_k[\![k_1]\!]\mathcal{D}_v[\![v_3]\!]\mathcal{D}_s[\![s]\!] \\
&= (\lambda\epsilon.\mathcal{B}_\tau[\![b_2]\!]e_2\rho_G\epsilon\mathcal{D}_a[\![a_2]\!]\mathcal{D}_u[\![u_2]\!]\mathcal{D}_k[\![k_2]\!])\mathcal{D}_v[\![v_3]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_2]\!]e_2\rho_G\mathcal{D}_v[\![v_3]\!]\mathcal{D}_a[\![a_2]\!]\mathcal{D}_u[\![u_2]\!]\mathcal{D}_k[\![k_2]\!]\mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 32: $R =$ Primitive Apply-Escape-Halt.

Let $\Sigma = \langle t, \langle \texttt{\%\%apply} \rangle :: b_1, v, a, u, k, s \rangle$,
where $a = \langle v_1 \rangle ^\frown a_1 {}^\frown \langle v_2 \rangle$ and $v_2 = \langle \text{ESCAPE HALT } l_1 \rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, \langle\rangle, v_2, a, u, k, s \rangle$.

$\mathcal{O}'[\![\Sigma']\!]$ is undefined since not $v' \notin \texttt{R}$. Thus $\mathcal{O}'[\![\Sigma]\!]$ is undefined, and so we do not have to consider this case.

Case 33: $R =$ Primitive Eqv.

Let $\Sigma = \langle t, \langle \texttt{\%\%eqv} \rangle :: b_1, v, \langle v_1 \;\, v_2 \rangle ^\frown a_1, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v', \langle\rangle, u, k, s \rangle$,
where $v' = ((v_1 = v_2) \rightarrow \mathit{true}, \mathit{false})$.

$$\begin{aligned}
\mathcal{D}[\![\Sigma]\!] &= \mathcal{B}_\tau[\![\langle \texttt{\%\%eqv} \rangle :: b_1]\!] e\rho_G \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho . \mathcal{Z}_\tau[\![\langle \texttt{\%\%eqv} \rangle]\!] e\rho (\mathcal{B}_\tau[\![b_1]\!] e\rho)) e\rho_G \\
&\quad\; \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
&= \mathcal{Z}_\tau[\![\langle \texttt{\%\%eqv} \rangle]\!] e\rho_G (\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \\
&\quad\; \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
&= (\lambda e\rho\pi\epsilon\epsilon^*\rho_R\psi . \#\epsilon^* \geq 2 \rightarrow \pi \\
&\quad\; (((\mathit{rev}\ \epsilon^*)0 = (\mathit{rev}\ \epsilon^*)1 \rightarrow \mathit{true}, \mathit{false}) \text{ in } \texttt{E}) \langle\rangle \rho_R \psi, \\
&\quad\; \mathit{wrong} \text{ ``bad arg count''}) \\
&\quad\; e\rho_G (\mathcal{B}_\tau[\![b_1]\!] e\rho_G) \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
&= (\#\mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!] \geq 2 \rightarrow \mathcal{B}_\tau[\![b_1]\!] e\rho_G \\
&\quad\; (((\mathit{rev}\ \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!])0 = \\
&\quad\; (\mathit{rev}\ \mathcal{D}_a[\![\langle v_1 \;\, v_2 \rangle ^\frown a_1]\!])1 \rightarrow \mathit{true}, \mathit{false}) \text{ in } \texttt{E}) \\
&\quad\; \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!], \mathit{wrong} \text{ ``bad arg count''}) \mathcal{D}_s[\![s]\!] \\
&= \mathcal{B}_\tau[\![b_1]\!] e\rho_G ((\mathcal{D}_v[\![v_1]\!] = \mathcal{D}_v[\![v_2]\!] \rightarrow \mathit{true}, \mathit{false}) \text{ in } \texttt{E}) \\
&\quad\; \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
&= \mathcal{D}[\![\Sigma']\!]
\end{aligned}$$

Case 34: $R =$ Primitive Add.

Let $\Sigma = \langle t, \langle \texttt{\%\%add} \rangle :: b_1, v, a, u, k, s \rangle$,
where $t = \langle \texttt{template}\ b\ e \rangle$.

Then $R(\Sigma) = \Sigma' = \langle t, b_1, v', \langle\rangle, u, k, s \rangle$,
where $v' = \textit{n-ary-sum}(a)$.

Notice that $\textit{sum\_vals}\ (\textit{rev}\ \epsilon^*) = \textit{sum\_vals}\ \epsilon^*$ since addition is commutative.

$$\begin{array}{rcl}
\mathcal{D}[\![\Sigma]\!] & = & \mathcal{B}_\tau[\![\langle \texttt{\%\%add} \rangle :: b_1]\!] e \rho_G \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & (\lambda e \rho . \mathcal{Z}_\tau[\![\langle \texttt{\%\%add} \rangle]\!] e \rho (\mathcal{B}_\tau[\![b_1]\!] e \rho)) e \rho_G \\
 & & \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & \mathcal{Z}_\tau[\![\langle \texttt{\%\%add} \rangle]\!] e \rho_G (\mathcal{B}_\tau[\![b_1]\!] e \rho_G) \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & (\lambda e \rho \pi \epsilon \epsilon^* \rho_R \psi . \pi (\textit{sum_vals}\ (\textit{rev}\ \epsilon^*)) \langle\rangle \rho_R \psi) \\
 & & e \rho_G (\mathcal{B}_\tau[\![b_1]\!] e \rho_G) \mathcal{D}_v[\![v]\!] \mathcal{D}_a[\![a]\!] \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & \mathcal{B}_\tau[\![b_1]\!] e \rho_G (\textit{sum_vals}\ \mathcal{D}_a[\![a]\!]) \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & \mathcal{B}_\tau[\![b_1]\!] e \rho_G (\textit{fix} (\lambda f \epsilon^* . \# \epsilon^* = 0 \rightarrow f(\epsilon^* \dagger 1) + ((\epsilon^* 0) | \mathtt{R})) \mathcal{D}_a[\![a]\!]) \\
 & & \langle\rangle \mathcal{D}_u[\![u]\!] \mathcal{D}_k[\![k]\!] \mathcal{D}_s[\![s]\!] \\
 & = & \mathcal{D}[\![\Sigma']\!]
\end{array}$$

7 Proof of Lemma 5.4

Let x be an ABC state value, argument stack, environment, continuation, or store. $\hat{x}$ is the result of replacing each $u = \langle \text{ENV}\ u'\ l^* \rangle$ occurring in x with $\langle \text{ENV}\ u'\ (\textit{rev}\ l^*) \rangle$.

Two ABC states $\Sigma = \langle t, b, v, a, u, k, s \rangle$ and $\Sigma' = \langle t', b', v', a', u', k', s' \rangle$ are *compatible* if they are both normal, $t' = t$, $b' = b$, $v' = \hat{v}$, $a' = \hat{a}$, $u' = \hat{u}$, $k' = \hat{k}$, and there is a permutation p of s such that:

(1) $s' = \widehat{p(s)}$.

(2) For each $l \in L_{\text{glo}}(\Sigma)$, $s'(l) = \widehat{s(l)}$.

(3) For each $l \in L_{\text{env}}(\Sigma)$ with $l = \textit{env-reference}(u_0, n_0, n_1)$ for some u_0 occurring in Σ and $n_0, n_1 \in \mathtt{N}$, $s'(l') = \widehat{s(l)}$ where $l' = \textit{env-reference}(\widehat{u_0}, n_0, n_1)$.

(4) For each $l \in L_{\text{mp}}(\Sigma)$, $s'(l) = \widehat{s(l)}$.

(5) For each $l \in L_{\mathrm{ip}}(\Sigma)$, $s'(l) = \widehat{s(l)}$ ($= s(l)$ since immutable pairs never contain environments).

Lemma 7.1 *Let Σ and Σ' be compatible* ABC *states, R be an* ABC *rule, and R' be the corresponding special rule.*

(1) R is applicable to Σ iff R' is applicable to Σ'.

(2) If R is applicable to Σ and R' is applicable to Σ', then $R(\Sigma)$ and $R'(\Sigma')$ are compatible.

Proof Let Σ and Σ' be compatible ABC states, R be an ABC rule, and R' be the corresponding special rule.

Case 1: $R = R'$, i.e., R is not Make Environment.
(1): Certainly, R is applicable to Σ iff R is applicable to Σ' provided:

(a) The domain conditions of R do not depend on the form of any environment in Σ or Σ'.

(b) The domain conditions of R do not depend on the value of the store at a location in $L_{\mathrm{env}}(\Sigma)$.

There is just one ABC rule (besides Make Environment) which violates (a) or (b): Local. The domain conditions of Local require that:

$$s(\textit{env-reference}(u, n_0, n_1)) \neq \text{UNDEFINED}.$$

Since Σ and Σ' are compatible, and hence satisfy clause 3 of the definition of compatible, Σ satisfies this condition iff Σ' satisfies it.

(2): Assume R is applicable to Σ and R' is applicable to Σ'. Then, by Lemma 5.1, $R(\Sigma)$ and $R'(\Sigma')$ are both normal. Certainly, $R(\Sigma)$ and $R'(\Sigma')$ are compatible provided:

(c) R does not create any new environments in Σ or Σ'.

(d) R does not modify the store at a location in $L_{\mathrm{env}}(\Sigma)$.

There is just one ABC rule (besides Make Environment) which violates (c) or (d): Set Local. Set Local modifies the value of the store at a location $l = \textit{env-reference}(u, n_0, n_1)$. Since Σ and Σ' are compatible, and hence satisfy clause 3 of the definition of compatible, $R(\Sigma)$ and $R'(\Sigma')$ also satisfy clause 3. Therefore, $R(\Sigma)$ and $R'(\Sigma')$ are compatible.

Case 2: R is Make Environment and R' is Alternate Make Environment.

(1): Since the domain conditions of R and R' are identical and do not depend on anything but the form of b, clearly, R is applicable to Σ iff R is applicable to Σ'.

(2): Assume R is applicable to Σ and R' is applicable to Σ'. Then, by Lemma 5.1, $R(\Sigma)$ and $R'(\Sigma')$ are both normal. Let $R(\Sigma)$ and $R'(\Sigma')$ have the form $\langle t, b, v, a, u, k, s^\frown a\rangle$ and $\langle t, b, v, a, u', k, s'^\frown(\mathit{rev}\ a)\rangle$, respectively. By the definition of *add-layer'*, $u' = \hat{u}$. Since Σ and Σ' are compatible, there is a permutation p of s such that $s' = \widehat{p(s)}$, and hence there is obviously an extension p' of p that permutes $s^\frown a$ such that $s'^\frown(\mathit{rev}\, a) = p'(\widehat{s^\frown a})$. The new locations in stores $s^\frown a$ and $s'^\frown(\mathit{rev}\ a)$ are members of $L_{\text{env}}(R(\Sigma))$. Since the values in the respective stores at these new locations are given in reverse order from one another, clause 3 of the definition of compatible holds. Clauses 2, 4, and 5 of the definition of compatible also hold since the new stores are extensions of the old stores and all of the new locations are members of $L_{\text{env}}(R(\Sigma))$. Therefore, $R(\Sigma)$ and $R'(\Sigma')$ are compatible. □

Proof of Lemma 5.4 Let t be a TBC template and Σ be an initial state for t, and assume $\mathcal{O}[\![\Sigma]\!]$ is defined. Since $\mathcal{O}[\![\Sigma]\!]$ is defined, there is a finite sequence $\Sigma_0, \ldots, \Sigma_n$ of ABC states and a finite sequence $R_1, \ldots, R_n$ of ABC rules such that:

(1) $\Sigma = \Sigma_0$ and $0 \leq n$.

(2) $R_{i+1}(\Sigma_i) = \Sigma_{i+1}$ for all i with $0 \leq i \leq n-1$.

(3) Σ_n has the form $\langle t_0, \langle\rangle, v, a, u, k, s\rangle$ with $\mathcal{D}_v[\![v]\!]|\mathtt{R}$ in $\mathtt{A} = \mathcal{O}[\![\Sigma]\!]$.

Σ is compatible with itself since it is an initial state and contains no environments. Thus, by Lemma 7.1, there is also a finite sequence $\Sigma'_0, \ldots, \Sigma'_n$ of ABC states and a finite sequence $R'_1, \ldots, R'_n$ of special rules such that:

(1) $\Sigma = \Sigma'_0$.

(2) $R'_{i+1}(\Sigma'_i) = \Sigma'_{i+1}$ for all i with $0 \leq i \leq n-1$.

(3) Σ_i and Σ'_i are compatible for all i with $0 \leq i \leq n$.

Since Σ_n and Σ'_n are compatible, $\Sigma'_n = \langle t_0, \langle\rangle, v, \hat{a}, \hat{u}, \hat{k}, \widehat{p(s)}\rangle$ for some permutation p of s. Therefore, $\mathcal{O}'[\![\Sigma]\!] = \mathcal{D}_v[\![v]\!]|\mathtt{R}$ in $\mathtt{A} = \mathcal{O}[\![\Sigma]\!]$. □

References

[1] J. D. Guttman, L. G. Monk, W. M. Farmer, J. D. Ramsdell, and V. Swarup. The VLISP byte-code compiler. M 92B092, The MITRE Corporation, 1992.

[2] J. D. Guttman, L. G. Monk, W. M. Farmer, J. D. Ramsdell, and V. Swarup. The VLISP flattener. M 92B094, The MITRE Corporation, 1992.