SNESL formalization Level-1

Dandan Xue

September 25, 2017

0 Level-1

Draft version 0.1.0:

- added using variables in comprehensions (grammar and rules updated, proof not yet)

1 Source Language

1.1 Source language syntax

SNESL Expressions:

$$e ::= x \mid \textbf{let } x = e_1 \textbf{ in } e_2 \mid \phi(x_1, ..., x_k) \mid \{e : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\}$$

$$\phi = \textbf{const}_n \mid \textbf{iota} \mid \textbf{plus}$$

SNESL values:

$$n \in \mathbf{Z}$$

$$v ::= n \mid \{v_1, ..., v_k\}$$

1.2 Type system

$$\tau ::= \textbf{int} | \{\tau_1\}$$

Type environment $\Gamma = [x_1 \mapsto \tau_1, ..., x_i \mapsto \tau_i]$.

- Expression typing rules:

  Judgment $\boxed{\Gamma \vdash e : \tau}$

  $$\frac{}{\Gamma \vdash x : \tau}\,(\Gamma(x) = \tau) \qquad \frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma[x \mapsto \tau_1] \vdash e_2 : \tau}{\Gamma \vdash \textbf{let } x = e_1 \textbf{ in } e_2 : \tau}$$

  $$\frac{\phi : (\tau_1, ..., \tau_k) \to \tau}{\Gamma \vdash \phi(x_1, ..., x_k) : \tau}\,((\Gamma(x_i) = \tau_i)_{i=1}^k)$$

  $$\frac{[x \mapsto \tau_1, (x_i \mapsto \textbf{int})_{i=1}^j] \vdash e : \tau}{\Gamma \vdash \{e : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\} : \{\tau\}}\,(\Gamma(y) = \{\tau_1\}, (\Gamma(x_i) = \textbf{int})_{i=1}^j)$$

- Auxiliary Judgment $\boxed{\phi : (\tau_1, ..., \tau_k) \to \tau}$

$$\frac{}{\mathbf{const}_n : () \to \mathbf{int}} \qquad \frac{}{\mathbf{iota} : (\mathbf{int}) \to \{\mathbf{int}\}} \qquad \frac{}{\mathbf{plus} : (\mathbf{int}, \mathbf{int}) \to \mathbf{int}}$$

- Value typing rules:

  Judgment $\boxed{v : \tau}$

$$\frac{}{n : \mathbf{int}} \qquad \frac{(v_i : \tau)_{i=1}^k}{\{v_1, ..., v_k\} : \{\tau\}}$$

1.3 Source language semantics

$\rho = [x_1 \mapsto v_1, ..., x_i \mapsto v_i]$

- Judgment $\boxed{\rho \ \vdash \ e \downarrow v}$

$$\frac{}{\rho \ \vdash \ x \downarrow v}\ (\rho(x) = v) \qquad \frac{\rho \ \vdash \ e_1 \downarrow v_1 \qquad \rho[x \mapsto v_1] \ \vdash \ e_2 \downarrow v}{\rho \ \vdash \ \mathbf{let}\ e_1 = x\ \mathbf{in}\ e_2 \downarrow v}$$

$$\frac{\phi(v_1, ..., v_k) \ \vdash \ v}{\rho \ \vdash \ \phi(x_1, ..., x_k) \downarrow v}\ ((\rho(x_i) = v_i)_{i=1}^k)$$

$$\frac{([x \mapsto v_i, (x_i \mapsto n_i)_{i=1}^j] \ \vdash \ e \downarrow v_i')_{i=1}^k}{\rho \ \vdash \ \{e : x\ \mathbf{in}\ y\ \mathbf{using}\ x_1, ..., x_j\} \downarrow \{v_1', ..., v_k'\}}\ (\rho(y) = \{v_1, ..., v_k\}, (\rho(x_i) = n_i)_{i=1}^j)$$

- Auxiliary Judgment $\boxed{\phi(v_1, ..., v_k) \ \vdash \ v}$

$$\frac{}{\mathbf{const}_n() \ \vdash \ n} \qquad \frac{}{\mathbf{iota}(n) \ \vdash \ \{0, 1, ..., n-1\}}\ (n \geq 0)$$

$$\frac{}{\mathbf{plus}(n_1, n_2) \ \vdash \ n_3}\ (n_3 = n_1 + n_2)$$

2 Target language

2.1 SVCODE syntax

(1) Stream id:

$$s \in \mathbf{SId} = \mathbf{N} = \{0, 1, 2...\}$$

A list of **SId**:

$$\mathbf{S} = [s_1, ..., s_i]^1$$

(2) SVCODE operations:

$$\psi ::= \texttt{Const}_\texttt{a} \mid \texttt{ToFlags} \mid \texttt{Usum} \mid \texttt{MapTwo}_\oplus \mid \texttt{ScanPlus}_{n_0} \mid \texttt{Distr}$$

where $\oplus$ stands for some binary operation on **int**.

[1] For simplicity, in some cases where duplicate of some elements does not affect the correctness we will also use **S** to stand for a set of **SId**s which keeps only one copy of each element in **S**.

(3) SVCODE program:

$$
\begin{aligned}
p ::= \ & \epsilon \\
& \mid s := \psi(s_1, ..., s_i) \\
& \mid \mathbf{S}_{out} := \texttt{WithCtrl}(s, \mathbf{S}_{in}, p_1) \qquad (\texttt{fv}(p_1) \subseteq \mathbf{S}_{in}, \mathbf{S}_{out} \subseteq \texttt{dv}(p_1)) \\
& \mid p_1; p_2
\end{aligned}
$$

Note: here this $\mathbf{S}_{in}$ is actually slightly different from the import list used in our streaming interpreter which does not contain the new control stream id. So it may cause some confusion later when we introduce the streaming language. We may want to remove this $\mathbf{S}_{in}$ from WithCtrl instruction if it turns out it is not necessary to make it an explicit component here.

(4) Differentce of sets:
For two sets A and B,

$$A - B = \{s | s \in A \wedge s \notin B\}$$

It is easy to prove the following properties:

- For any three sets A, B and C:

$$(A - B) \cap C = (A \cap C) - B = A \cap (C - B) \tag{2.1}$$

- For two sets A and B,

$$A \cap B = \emptyset \Leftrightarrow A - B = A \tag{2.2}$$

(5) Defined variables: a set (or list) of stream ids that are defined by a SVCODE progrom and accessible to the outside environment of the program

$$
\begin{aligned}
&\texttt{dv}(\epsilon) \ = \{\} \\
&\texttt{dv}(s := \psi(s_1, ..., s_i)) = \{s\} \\
&\texttt{dv}(\mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_1)) = \mathbf{S}_{out} \\
&\texttt{dv}(p_1; p_2) = \texttt{dv}(p_1) \cup \texttt{dv}(p_2)
\end{aligned}
$$

(6) Free variables: a set (or list) of stream ids that are not defined but referred to by a SVCODE program

$$
\begin{aligned}
&\texttt{fv}(\epsilon) \ = \{\} \\
&\texttt{fv}(s := \psi(s_1, ..., s_i)) = \{s_1, ..., s_i\} \\
&\texttt{fv}(\mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_1)) = \{s_c\} \cup \mathbf{S}_{in} \\
&\texttt{fv}(p_1; p_2) = \texttt{fv}(p_1) \cup \texttt{fv}(p_2) - \texttt{dv}(p_1)
\end{aligned}
$$

(7) SVCODE streams:

$$
\begin{aligned}
b &\in \{\texttt{T}, \texttt{F}\} \\
a &::= n \mid b \mid () \\
\vec{b} &= \langle b_1, ..., b_i \rangle \\
\vec{a} &= \langle a_1, ..., a_i \rangle
\end{aligned}
$$

(8) Notations and operations about streams:

- Let $\langle a_0|...\rangle$ denote a non-empty stream with the first element a_0, and $\langle a_0|\vec{a}\rangle$ also a non-empty stream $\langle a_0, a_1, ..., a_i\rangle$ for some $\vec{a} = \langle a_1, ..., a_i\rangle$.
- $\langle a_1, ..., a_i\rangle +\!\!+ \langle a'_1, ..., a'_j\rangle = \langle a_1, ..., a_i, a'_1, ..., a'_j\rangle$

2.2 SVCODE semantics

SVCODE stores $\sigma = [s_1 \mapsto \vec{a}_1, ..., s_i \mapsto \vec{a}_i]$.

- Judgment $\boxed{\langle p, \sigma\rangle \downarrow^{\vec{c}} \sigma'}$

  $\vec{c}$ is the control stream.

  P-Empty : $\dfrac{}{\langle \epsilon, \sigma\rangle \downarrow^{\vec{c}} \sigma}$

  P-Xducer : $\dfrac{\psi(\vec{a}_1, ..., \vec{a}_k) \downarrow^{\vec{c}} \vec{a}}{\langle s := \psi(s_1, ..., s_k), \sigma\rangle \downarrow^{\vec{c}} \sigma[s \mapsto \vec{a}]}\ ((\sigma(s_i) = \vec{a}_i)_{i=1}^k)$

  P-Wc-Emp : $\dfrac{}{\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_1), \sigma\rangle \downarrow^{\vec{c}} \sigma[(s_i \mapsto \langle\rangle)_{i=1}^l]}\ \begin{pmatrix} \forall s \in \{s_c\} \cup \mathbf{S}_{in}.\sigma(s) = \langle\rangle \\ \mathbf{S}_{out} = [s_1, ..., s_l] \end{pmatrix}$

  P-Wc-Nonemp : $\dfrac{\langle p_1, \sigma\rangle \downarrow^{\vec{c}_1} \sigma''}{\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_1), \sigma\rangle \downarrow^{\vec{c}} \sigma[(s_i \mapsto \sigma''(s_i))_{i=1}^l]}\ \begin{pmatrix} \sigma(s_c) = \vec{c}_1 = \langle ()|...\rangle \\ \mathbf{S}_{out} = [s_1, ..., s_l] \end{pmatrix}$

  P-Seq : $\dfrac{\langle p_1, \sigma\rangle \downarrow^{\vec{c}} \sigma'' \qquad \langle p_2, \sigma''\rangle \downarrow^{\vec{c}} \sigma'}{\langle p_1; p_2, \sigma\rangle \downarrow^{\vec{c}} \sigma'}$

- *Transducer* semantics:

  Judgment $\boxed{\psi(\vec{a}_1, ..., \vec{a}_k) \downarrow^{\vec{c}} \vec{a}}$

  P-X-Loop : $\dfrac{\psi(\vec{a}_{11}, ..., \vec{a}_{k1}) \Downarrow \vec{a}_1 \qquad \psi(\vec{a}_{12}, ..., \vec{a}_{k2}) \downarrow^{\vec{c}} \vec{a}_2}{\psi(\vec{a}_{11} +\!\!+ \vec{a}_{12}, ..., \vec{a}_{k1} +\!\!+ \vec{a}_{k2}) \downarrow^{\langle a_0|\vec{c}\rangle} \vec{a}}\ (\vec{a} = \vec{a}_1 +\!\!+ \vec{a}_2)$

  P-X-Termi : $\dfrac{}{\psi(\langle\rangle_1, ..., \langle\rangle_k) \downarrow^{\langle\rangle} \langle\rangle}$ [2]

- Transducer *block* semantics:

  Judgment $\boxed{\psi(\vec{a}_1, ..., \vec{a}_k) \Downarrow \vec{a}}$

  P-Const : $\dfrac{}{\texttt{Const}_{\texttt{a}}() \Downarrow \langle a\rangle}$ P-ToFlags : $\dfrac{}{\texttt{ToFlags}(\langle n\rangle) \Downarrow \langle \texttt{F}_1, ..., \texttt{F}_n, \texttt{T}\rangle}$

  P-MapTwo : $\dfrac{}{\texttt{MapTwo}_{\oplus}(\langle n_1\rangle, \langle n_2\rangle) \Downarrow \langle n_3\rangle}\ (n_3 = n_1 \oplus n_2)$

[2] For convenience, in this thesis we add subscripts to a sequence of constants, such as $\langle\rangle, \texttt{F}, 1$, to denote the total number of these constants.

$$\text{P-UsumF}: \frac{\texttt{Usum}(\vec{b}) \Downarrow \vec{a}}{\texttt{Usum}(\langle \texttt{F}|\vec{b}\rangle) \Downarrow \langle()|\vec{a}\rangle} \qquad \text{P-UsumT}: \frac{}{\texttt{Usum}(\langle \texttt{T}\rangle) \Downarrow \langle\rangle}$$

$$\text{P-ScanF}: \frac{\texttt{ScanPlus}_{n_0+n}(\vec{b}, \vec{a}) \Downarrow \vec{a}'}{\texttt{ScanPlus}_{n_0}(\langle \texttt{F}|\vec{b}\rangle, \langle n|\vec{a}\rangle) \Downarrow \langle n_0|\vec{a}'\rangle} \qquad \text{P-ScanT}: \frac{}{\texttt{ScanPlus}_{n_0}(\langle \texttt{T}\rangle, \langle\rangle) \Downarrow \langle\rangle}$$

$$\text{P-DistrF}: \frac{\texttt{Distr}(\vec{b}, \langle n\rangle) \Downarrow \vec{a}}{\texttt{Distr}(\langle \texttt{F}|\vec{b}\rangle, \langle n\rangle) \Downarrow \langle n|\vec{a}\rangle} \qquad \text{P-DistrT}: \frac{}{\texttt{Distr}(\langle \texttt{T}\rangle, \langle n\rangle) \Downarrow \langle\rangle}$$

Or if we want to use *unary* semantics maybe for later:

$$\frac{\psi(\langle \texttt{F}\rangle, ..., \vec{a}_{k1}) \downdownarrows \vec{a}_1 \qquad \psi(\vec{a}_{12}, ..., \vec{a}_{k2}) \Downarrow \vec{a}_2}{\psi(\langle \texttt{F}\rangle{+\!+}\vec{a}_{12}, ..., \vec{a}_{k1}{+\!+}\vec{a}_{k2}) \Downarrow \vec{a}} \, (\vec{a} = \vec{a}_1{+\!+}\vec{a}_2)$$

$$\frac{\psi(\langle \texttt{T}\rangle, ..., \vec{a}_k) \downdownarrows \vec{a}}{\psi(\langle \texttt{T}\rangle, ..., \vec{a}_k) \Downarrow \vec{a}}$$

– Transducer *unary* semantics:

Judgment $\boxed{\psi(\langle b\rangle, ..., \vec{a}_k) \downdownarrows \vec{a}}$

$$\frac{}{\texttt{Usum}(\langle \texttt{F}\rangle) \downdownarrows \langle()\rangle} \qquad \frac{}{\texttt{Usum}(\langle \texttt{T}\rangle) \downdownarrows \langle\rangle}$$

– Transducer block with *accumulator*:

Judgment $\boxed{\psi_n(\vec{a}_1, ..., \vec{a}_k) \Downarrow \vec{a}}$

$$\frac{\psi_{n_0}(\langle \texttt{F}\rangle, ..., \vec{a}_{k1}) \downdownarrows^{n'_0} \langle n_1\rangle \qquad \psi_{n'_0}(\vec{a}_{12}, ..., \vec{a}_{k2}) \Downarrow \vec{a}_2}{\psi_{n_0}(\langle \texttt{F}\rangle{+\!+}\vec{a}_{12}, ..., \vec{a}_{k1}{+\!+}\vec{a}_{k2}) \Downarrow \langle n_1\rangle{+\!+}\vec{a}_2}$$

$$\frac{\psi_{n_0}(\langle \texttt{T}\rangle, ..., \vec{a}_k) \downdownarrows \vec{a}}{\psi_{n_0}(\langle \texttt{T}\rangle, ..., \vec{a}_k) \Downarrow \vec{a}}$$

– Transducer unary with *accumulator*:

Judgment $\boxed{\psi_n(\langle \texttt{F}\rangle, ..., \vec{a}_k) \downdownarrows^{n'} \vec{a}}$

$$\frac{}{\texttt{ScanPlus}_{n_0}(\langle \texttt{F}\rangle, \langle n\rangle) \downdownarrows^{n_0+n} \langle n_0\rangle}$$

Judgment $\boxed{\psi_n(\langle \texttt{T}\rangle, ..., \vec{a}_k) \downdownarrows \vec{a}}$

$$\frac{}{\texttt{ScanPlus}_{n_0}(\langle \texttt{T}\rangle, \langle\rangle) \downdownarrows \langle\rangle}$$

2.3 Definitions

We first define a binary relation $\overset{\mathbf{S}}{\sim}$ on stores to denote that two stores are *similar*: they have identical domains, and their bound values by $\mathbf{S}$ are the same. We call this $\mathbf{S}$ an *overlap* of these two stores.

Definition 2.1 (**Stores similarity**). $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$ *iff*
(1) $dom(\sigma_1) = dom(\sigma_2)$
(2) $\forall s \in \mathbf{S}.\sigma_1(s) = \sigma_2(s)$

According to this definition, it is only meaningful to have $\mathbf{S} \subseteq dom(\sigma_1)$ $(= dom(\sigma_2))$. When $\mathbf{S} = dom(\sigma_1) = dom(\sigma_2)$, σ_1 and σ_2 are identical. It is easy to show that this relation $\overset{\mathbf{S}}{\sim}$ is symmetric and transitive.

- If $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$, then $\sigma_2 \overset{\mathbf{S}}{\sim} \sigma_1$.
- If $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$ and $\sigma_2 \overset{\mathbf{S}}{\sim} \sigma_3$, then $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_3$.

We define another binary operation $\overset{\mathbf{S}}{\bowtie}$ on stores to denote a kind of specical concatenation of two similar stores: the *concatenation* of two similar stores is a new store, in which the bound values by $\mathbf{S}$ are from any of the parameter stores, and the others are the concatenation of the values from the two stores. In other words, a *concatenation* of two similar stores is only a concatenation of the bound values that *maybe* different in these stores.

Definition 2.2 (**Store Concatenation**). $\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 = \sigma$ *iff*
(1) $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$
(2) $\sigma(s) = \begin{cases} \sigma_i(s), & s \in \mathbf{S}, i \in \{1,2\} \\ \sigma_1(s) +\!\!+ \sigma_2(s), & otherwise \end{cases}$

Lemma 2.3. *If* $\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 = \sigma$, *then* $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma$ *and* $\sigma_2 \overset{\mathbf{S}}{\sim} \sigma$.

This lemma says that the concatenation result of two similar stores is still similar to each of them.

Lemma 2.4. *If* $\psi(\vec{a}_{11}, ..., \vec{a}_{1k}) \downarrow^{\vec{c}_1} \vec{a}_1$, *and* $\psi(\vec{a}_{21}, ..., \vec{a}_{2k}) \downarrow^{\vec{c}_2} \vec{a}_2$, *then* $\psi(\vec{a}_{11} +\!\!+ \vec{a}_{21}, ..., \vec{a}_{1k} +\!\!+ \vec{a}_{2k}) \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \vec{a}_1 +\!\!+ \vec{a}_2$.

Lemma 2.5. *If*

(i) $\sigma_1 \overset{S}{\sim} \sigma_2$

(ii) $\langle p, \sigma_1 \rangle \downarrow^{\vec{c}} \sigma$

(iii) $\mathtt{fv}(p) \cap \mathbf{S} = \emptyset$

(iv) $\forall s \in \mathtt{fv}(p).\sigma_2(s) = \langle\rangle$

then

(v) $\left\langle p, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}} \sigma'$

(vi) $\forall s' \in \mathtt{dv}(p).\sigma(s') = \sigma'(s')$

Lemma 2.6 (**Stores concatenation lemma**). *If*

(i) $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$

(ii) $\langle p, \sigma_1 \rangle \downarrow^{\vec{c}_1} \sigma_1'$ *(by some derivation* $\mathcal{P}_1$*)*

(iii) $\langle p, \sigma_2 \rangle \downarrow^{\vec{c}_2} \sigma_2'$ *(by some derivation* $\mathcal{P}_2$*)*

(iv) $\mathtt{fv}(p) \cap \mathbf{S} = \emptyset$

then $\left\langle p, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1 +\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$ *(by* $\mathcal{P}$*).*

We need this lemma to prove that the results of single computations inside a comprehension body (i.e. p in the lemma) can be concatenated to express a parallel computation. From the other direction, we can consider this process as distributing or splitting the compuation p on even smaller degree of parallel computations, in which all the supplier streams, i.e., $\mathtt{fv}(p)$, are splitted to feed the transducers. The splitted parallel degrees are specified by the control streams, i.e., $\vec{c}_1$ and $\vec{c}_2$ in the lemma. Other untouched **SId**s in all σs (i.e., $\mathbf{S}$) have no change throughout the process.

Proof. By induction on the syntax of p.

- Case $p = \epsilon$.
  $\mathcal{P}_1$ must be $\overline{\langle \epsilon, \sigma_1 \rangle \downarrow^{\vec{c}_1} \sigma_1}$, and $\mathcal{P}_2$ must be $\overline{\langle \epsilon, \sigma_2 \rangle \downarrow^{\vec{c}_2} \sigma_2}$.
  So $\sigma_1' = \sigma_1$, and $\sigma_2' = \sigma_2$, thus $\sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2' = \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2$.

  By P-Empty, we take $\mathcal{P} = \overline{\left\langle \epsilon, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1 +\!+ \vec{c}_2} \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2}$ and we are done.

- Case $p = s_l := \psi(s_1, ..., s_i); p'$.
  $\mathcal{P}_1$ must look like

$$\frac{\mathcal{P}_{10} \qquad \mathcal{P}_{11}}{\psi(\vec{a}_1, ..., \vec{a}_k) \downarrow^{\vec{c}_1} \vec{a} \qquad \langle p', \sigma_1[s_l \mapsto \vec{a}] \rangle \downarrow^{\vec{c}_1} \sigma_1'}{\langle s_l := \psi(s_1, ..., s_k); p', \sigma_1 \rangle \downarrow^{\vec{c}_1} \sigma_1'}$$

  and we have

$$(\sigma_1(s_i) = \vec{a}_i)_{i=1}^{k} \tag{2.3}$$

  Similarly, $\mathcal{P}_2$ must look like

$$\frac{\mathcal{P}_{20} \qquad \mathcal{P}_{21}}{\psi(\vec{a}_1', ..., \vec{a}_k') \downarrow^{\vec{c}_2} \vec{a}' \qquad \langle p', \sigma_2[s_l \mapsto \vec{a}'] \rangle \downarrow^{\vec{c}_2} \sigma_2'}{\langle s_l := \psi(s_1, ..., s_k); p', \sigma_2 \rangle \downarrow^{\vec{c}_2} \sigma_2'}$$

  and we have

$$(\sigma_2(s_i) = \vec{a}_i')_{i=1}^{k} \tag{2.4}$$

  From assumption (iv) we have

$$\begin{aligned}
\mathtt{fv}(s_l := \psi(s_1, ..., s_k); p') \cap \mathbf{S} &= \emptyset \\
(\{s_1, ..., s_k\} \cup \mathtt{fv}(p') - \{s_l\}) \cap \mathbf{S} &= \emptyset && \text{(by definition of } \mathtt{fv}()) \\
(\{s_1, ..., s_k\} \cup \mathtt{fv}(p')) \cap (\mathbf{S} - \{s_l\}) &= \emptyset && \text{(by (2.1))} \\
(\{s_1, ..., s_k\} \cup \mathtt{fv}(p')) \cap \mathbf{S} &= \emptyset && \text{(by (2.2) with ??? } \mathbf{S} \cap \{s_l\} = \emptyset) \\
(\{s_1, ..., s_k\} \cap \mathbf{S}) \cup (\mathtt{fv}(p') \cap \mathbf{S}) &= \emptyset
\end{aligned}$$

thus

$$\{s_1, ..., s_k\} \cap \mathbf{S} = \emptyset \tag{2.5}$$

$$\mathtt{fv}(p') \cap \mathbf{S} = \emptyset \tag{2.6}$$

By Lemma 2.4 on $\mathcal{P}_{10}$, $\mathcal{P}_{20}$, we get a derivation $\mathcal{P}'$ of

$$\psi(\vec{a}_1{+}{+}\vec{a}'_1, ..., \vec{a}_k{+}{+}\vec{a}'_k) \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \vec{a}{+}{+}\vec{a}'$$

Since $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$, by the definitons of store similarity and concatenation, it is easy to show that

$$\sigma_1[s_l \mapsto \vec{a}] \overset{\mathbf{S}}{\sim} \sigma_2[s_l \mapsto \vec{a}'] \tag{2.7}$$

and

$$\sigma_1[s_l \mapsto \vec{a}] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_l \mapsto \vec{a}'] = \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_l \mapsto \vec{a}{+}{+}\vec{a}'] \tag{2.8}$$

Then by IH on (2.7), $\mathcal{P}_{11}$, $\mathcal{P}_{21}$, (2.6), we obtain a derivation $\mathcal{P}''$ of

$$\left\langle p', \sigma_1[s_l \mapsto \vec{a}] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_l \mapsto \vec{a}'] \right\rangle \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \sigma'_1 \overset{\mathbf{S}}{\bowtie} \sigma'_2$$

By replacing the start store in $\mathcal{P}''$ with the right-hand side of (2.8), we get $\mathcal{P}'''$ of

$$\left\langle p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_l \mapsto \vec{a}{+}{+}\vec{a}'] \right\rangle \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \sigma'_1 \overset{\mathbf{S}}{\bowtie} \sigma'_2$$

With (2.3),(2.4) and (2.5), by the definition of store concatenation,

$$\forall i \in \{1, ..., k\}. \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s_i) = \sigma_1(s_i){+}{+}\sigma_2(s_i) = \vec{a}_i{+}{+}\vec{a}'_i,$$

Therefore using the rule P-XDUCER, we can build $\mathcal{P}$ as follows

$$\frac{\overset{\mathcal{P}'}{\psi(\vec{a}_1{+}{+}\vec{a}'_1, ..., \vec{a}_k{+}{+}\vec{a}'_k) \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \vec{a}{+}{+}\vec{a}'} \qquad \overset{\mathcal{P}'''}{\left\langle p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_l \mapsto \vec{a}{+}{+}\vec{a}'] \right\rangle \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \sigma'_1 \overset{\mathbf{S}}{\bowtie} \sigma'_2}}{\left\langle s_l := \psi(s_1, ..., s_k); p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1{+}{+}\vec{c}_2} \sigma'_1 \overset{\mathbf{S}}{\bowtie} \sigma'_2}$$

as required.

- Case $p = \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p'$ where

$$\mathbf{S}_{in} = \{s_c\} \cup \mathtt{fv}(p_0) \tag{2.9}$$

and

$$\mathbf{S}_{out} \subseteq \mathtt{dv}(p_0) \tag{2.10}$$

From the assumption (iv), we have

$$\begin{aligned}
&\mathtt{fv}(\mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p') \cap \mathbf{S} = \emptyset \\
&(\mathbf{S}_{in} \cup \mathtt{fv}(p') - \mathbf{S}_{out}) \cap \mathbf{S} = \emptyset && \text{(by definition of } \mathtt{fv}()) \\
&(\mathbf{S}_{in} \cup \mathtt{fv}(p')) \cap (\mathbf{S} - \mathbf{S}_{out}) = \emptyset && \text{(by (2.1))} \\
&(\mathbf{S}_{in} \cup \mathtt{fv}(p')) \cap \mathbf{S} = \emptyset && \text{(by (2.2) with ??? } \mathbf{S} \cap \mathbf{S}_{out} = \emptyset) \\
&(\mathbf{S}_{in} \cap \mathbf{S}) \cup (\mathtt{fv}(p') \cap \mathbf{S}) = \emptyset
\end{aligned}$$

thus

$$\mathbf{S}_{in} \cap \mathbf{S} = \emptyset \tag{2.11}$$

$$\mathtt{fv}(p') \cap \mathbf{S} = \emptyset \tag{2.12}$$

Since (2.9) with (2.11), we also have

$$\{s_c\} \cap \mathbf{S} = \emptyset \tag{2.13}$$

$$\mathtt{fv}(p_0) \cap \mathbf{S} = \emptyset \tag{2.14}$$

Assume $\mathbf{S}_{out} = [s_1, ..., s_j]$.
There are four possibilities:

– Subcase both $\mathcal{P}_1$ and $\mathcal{P}_2$ use P-Wc-Emp.

So $\mathcal{P}_1$ must look like

$$\frac{\begin{array}{c}\mathcal{P}_1' \\ \langle p', \sigma_1[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle]\rangle \downarrow^{\vec{c}_1} \sigma_1'\end{array}}{\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1\rangle \downarrow^{\vec{c}_1} \sigma_1'}$$

and we have

$$\forall s \in \mathbf{S}_{in}.\sigma_1(s) = \langle\rangle$$

that is,

$$\sigma_1(s_c) = \langle\rangle \tag{2.15}$$

and

$$\forall s \in \mathtt{fv}(p_0).\sigma_1(s) = \langle\rangle \tag{2.16}$$

Similarly,$\mathcal{P}_2$ must look like

$$\frac{\begin{array}{c}\mathcal{P}_2' \\ \langle p', \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle]\rangle \downarrow^{\vec{c}_2} \sigma_2'\end{array}}{\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_2\rangle \downarrow^{\vec{c}_2} \sigma_2'}$$

and we have

$$\forall s \in \mathbf{S}_{in}.\sigma_2(s) = \langle\rangle$$

that is,

$$\sigma_2(s_c) = \langle\rangle \tag{2.17}$$

and

$$\forall s \in \mathtt{fv}(p_0).\sigma_2(s) = \langle\rangle \tag{2.18}$$

Since $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$, it is easy to show that

$$\sigma_1[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \overset{\mathbf{S}}{\sim} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \tag{2.19}$$

and

$$\sigma_1[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] = (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \tag{2.20}$$

By IH on (2.19) with $\mathcal{P}_1'$, $\mathcal{P}_2'$, (2.12), we get a derivation

$$\left\langle p', \sigma_1[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2',$$

and replacing the start store with the right-hand side of (2.20) gives us a derivation $\mathcal{P}'$ of

$$\left\langle p', (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$$

Since (2.13), by the definition of store concatenation with (2.15), (2.17), we have

$$\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s_c) = \sigma_1(s_c) +\!\!+ \sigma_2(s_c) = \langle\rangle$$

Also, with (2.16) and (2.18), we must have

$$\forall s \in \mathtt{fv}(p_0).\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s) = \langle\rangle$$

Thus, $\forall s \in \mathbf{S}_{in}.\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s) = \langle\rangle$.
Therefore, using P-Wc-Emp we can build $\mathcal{P}$ as follows

$$\frac{\begin{array}{c}\mathcal{P}' \\ \left\langle p', (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'\end{array}}{\left\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'}$$

and we are done.

– Subcase $\mathcal{P}_1$ uses P-Wc-Nomemp, $\mathcal{P}_2$ uses P-Wc-Emp.

$\mathcal{P}_1$ must look like

$$\frac{\begin{array}{cc}\mathcal{P}_1' & \mathcal{P}_1'' \\ \langle p_0, \sigma_1\rangle \downarrow^{\vec{c}_1'} \sigma_1'' & \langle p', \sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma''(s_j)]\rangle \downarrow^{\vec{c}_1} \sigma_1'\end{array}}{\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1\rangle \downarrow^{\vec{c}_1} \sigma_1'}$$

and we have

$$\sigma_1(s_c) = \vec{c}_1' = \langle ()|...\rangle \tag{2.21}$$

$\mathcal{P}_2$ must look like

$$\frac{\begin{array}{c}\mathcal{P}_2' \\ \langle p', \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle]\rangle \downarrow^{\vec{c}_2} \sigma_2'\end{array}}{\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_2\rangle \downarrow^{\vec{c}_2} \sigma_2'}$$

and we have

$$\sigma_2(s_c) = \langle\rangle \tag{2.22}$$

and

$$\forall s \in \mathtt{fv}(p_0).\sigma_2(s) = \langle\rangle \tag{2.23}$$

By Lemma 2.5 on $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$ with $\mathcal{P}_1'$, (2.23), (2.14), we obtain a derivation $\mathcal{P}_0$ of

$$\left\langle p_0, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1} \sigma_0$$

for some σ_0, and

$$\forall s \in dv(p_0).\sigma_0(s) = \sigma_1''(s),$$

thus, with (2.10), we have

$$(\sigma_0(s_i) = \sigma_1''(s_i))_{i=1}^{j} \tag{2.24}$$

Since $\sigma_1 \overset{\mathbf{S}}{\sim} \sigma_2$, it is easy to show that

$$\sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\sim} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \tag{2.25}$$

and

$$\sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] = (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s \tag{2.26}$$

By IH on (2.25) with $\mathcal{P}_1''$, $\mathcal{P}_2'$, (2.12), we get a derivation

$$\left\langle p', \sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \langle\rangle, ..., s_j \mapsto \langle\rangle] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2',$$

and replacing the start store with the right-hand side of (2.26) gives us a derivation $\mathcal{P}'$ of

$$\left\langle p', (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$$

With (2.24), we replcace $\sigma_1''(i)$ with $\sigma_0(i)$ for all $i \in \{1, ..., j\}$, then we obtain $\mathcal{P}''$ of

$$\left\langle p', (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \sigma_0(s_1), ..., s_j \mapsto \sigma_0(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$$

By the definition of store concatenation with (2.21), (2.22), we have

$$\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s_c) = \sigma_1(s_c) +\!\!+ \sigma_2(s_c) = \vec{c}_1' = \langle ()|... \rangle$$

Therefore, we use the rule P-Wc-Nonemp to build $\mathcal{P}$ as follows

$$\frac{\begin{array}{cc} \mathcal{P}_0 & \mathcal{P}'' \\ \left\langle p_0, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1} \sigma_0 & \left\langle p', (\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2)[s_1 \mapsto \sigma_0(s_1), ..., s_j \mapsto \sigma_0(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2' \end{array}}{\left\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'}$$

and we are done.

– Subcase $\mathcal{P}_1$ uses P-Wc-Emp and $\mathcal{P}_2$ uses P-Wc-Nonemp.
This subcase is symmetric to the second one.

– Subcase both $\mathcal{P}_1$ and $\mathcal{P}_2$ use P-Wc-Nonemp.
$\mathcal{P}_1$ must look like

$$\frac{\begin{array}{cc}\mathcal{P}_1' & \mathcal{P}_1'' \\ \langle p_0, \sigma_1\rangle \downarrow^{\vec{c}_1'} \sigma_1'' & \langle p', \sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma''(s_j)]\rangle \downarrow^{\vec{c}_1} \sigma_1'\end{array}}{\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1\rangle \downarrow^{\vec{c}_1} \sigma_1'}$$

and

$$\sigma_1(s_c) = \vec{c}_1' = \langle ()|...\rangle \tag{2.27}$$

Similarly, $\mathcal{P}_2$ must look like

$$\frac{\begin{array}{cc}\mathcal{P}_2' & \mathcal{P}_2'' \\ \langle p_0, \sigma_2\rangle \downarrow^{\vec{c}_2'} \sigma_2'' & \langle p', \sigma_2[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma''(s_j)]\rangle \downarrow^{\vec{c}_2} \sigma_2'\end{array}}{\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_2\rangle \downarrow^{\vec{c}_2} \sigma_2'}$$

and

$$\sigma_2(s_c) = \vec{c}_2' = \langle ()|...\rangle \tag{2.28}$$

By IH on $\mathcal{P}_1'$, $\mathcal{P}_2'$ with (2.14), we get a derivation $\mathcal{P}_0$ of

$$\left\langle p_0, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1' +\!+ \vec{c}_2'} \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''$$

Since (from $\mathbf{S} \cap \mathbf{S}_{out} = \emptyset$) $\forall i \in \{1, ..., j\}.s_i \notin \mathbf{S}$, then

$$\sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_i) = \sigma_1''(s_i) +\!+ \sigma_2''(s_i) \tag{2.29}$$

Also it is easy to show

$$\sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\sim} \sigma_2[s_1 \mapsto \sigma_2''(s_1), ..., s_j \mapsto \sigma_2''(s_j)] \tag{2.30}$$

and

$$\begin{aligned}\sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_2''(s_1), ..., s_j \mapsto \sigma_2''(s_j)] = \\ \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_1''(s_1) +\!+ \sigma_2''(s_1), ..., s_j \mapsto \sigma_1''(s_j) +\!+ \sigma_2''(s_j)]\end{aligned} \tag{2.31}$$

By IH on (2.30) with $\mathcal{P}_1''$, $\mathcal{P}_2''$, (2.12), we obtain a derivation of

$$\left\langle p', \sigma_1[s_1 \mapsto \sigma_1''(s_1), ..., s_j \mapsto \sigma_1''(s_j)] \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_2''(s_1), ..., s_j \mapsto \sigma_2''(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2',$$

and replacing the start store with the right-hand side of (2.31) gives us a derivation $\mathcal{P}'$ of

$$\left\langle p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_1''(s_1) +\!+ \sigma_2''(s_1), ..., s_j \mapsto \sigma_1''(s_j) +\!+ \sigma_2''(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$$

With (2.29), we can replace all the new bound values $\sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_i)$ in $\mathcal{P}'$ with $\sigma_1''(s_i) +\!+ \sigma_2''(s_i)$ for all $i \in \{1, ..., j\}$, giving us another derivation $\mathcal{P}''$ of

$$\left\langle p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_1), ..., s_j \mapsto \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'$$

Since (2.27) and (2.28), we know $\sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2(s_c) = \vec{c}_1' +\!\!+ \vec{c}_2' = \langle ()|... \rangle$, therefore we can use the rule P-Wc-Nonemp to build $\mathcal{P}$ as follows:

$$\frac{\mathcal{P}_0 \qquad\qquad \mathcal{P}''}{\left\langle p_0, \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1' +\!\!+ \vec{c}_2'} \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2'' \qquad \left\langle p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2[s_1 \mapsto \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_1), ..., s_j \mapsto \sigma_1'' \overset{\mathbf{S}}{\bowtie} \sigma_2''(s_j)] \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}}}{\left\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_c, \mathbf{S}_{in}, p_0); p', \sigma_1 \overset{\mathbf{S}}{\bowtie} \sigma_2 \right\rangle \downarrow^{\vec{c}_1 +\!\!+ \vec{c}_2} \sigma_1' \overset{\mathbf{S}}{\bowtie} \sigma_2'}$$

and we are done.

□

Let $\sigma_1 \overset{<s}{=\!=} \sigma_2$ denote $\forall s' < s. \sigma_1(s') = \sigma_2(s')$.

Lemma 2.7. *If* $\sigma_1 \overset{\mathbf{S}_1}{\sim} \sigma_1'$, $\sigma_2 \overset{\mathbf{S}_2}{\sim} \sigma_2'$, $\sigma_1 \overset{<s}{=\!=} \sigma_2$, *and* $\sigma_1' \overset{<s}{=\!=} \sigma_2'$ *then* $\sigma_1 \overset{\mathbf{S}_1}{\bowtie} \sigma_1' \overset{<s}{=\!=} \sigma_2 \overset{\mathbf{S}_2}{\bowtie} \sigma_2'$.

2.4 SVCODE determinism theroem

Definition 2.8. $\vec{a}$ *is a prefix of* $\vec{a}'$ *if* $\vec{a} \sqsubseteq \vec{a}'$*:*
Judgment $\boxed{\vec{a} \sqsubseteq \vec{a}'}$

$$\frac{}{\langle\rangle \sqsubseteq \vec{a}} \qquad \frac{\vec{a} \sqsubseteq \vec{a}'}{\langle a_0|\vec{a}\rangle \sqsubseteq \langle a_0|\vec{a}'\rangle}$$

Lemma 2.9. *If*

(i) $(\vec{a}_i' \sqsubseteq \vec{a}_i)_{i=1}^k$ *and* $\psi(\vec{a}_1', ..., \vec{a}_k') \Downarrow \vec{a}'$,

(ii) $(\vec{a}_i'' \sqsubseteq \vec{a}_i)_{i=1}^k$ *and* $\psi(\vec{a}_1'', ..., \vec{a}_k'') \Downarrow \vec{a}''$

then

(i) $(\vec{a}_i' = \vec{a}_i'')_{i=1}^k$

(ii) $\vec{a}' = \vec{a}''$.

Lemma 2.10. *If* $\psi(\vec{a}_1, ..., \vec{a}_k) \downarrow^{\vec{c}} \vec{a}$, *and* $\psi(\vec{a}_1, ..., \vec{a}_k) \downarrow^{\vec{c}} \vec{a}'$, *then* $\vec{a} = \vec{a}'$.

Theorem 2.11 (SVCODE determinism). *If* $\langle p, \sigma\rangle \downarrow^{\vec{c}} \sigma'$ *and* $\langle p, \sigma\rangle \downarrow^{\vec{c}} \sigma''$, *then* $\sigma' = \sigma''$.

3 Translation

3.1 Translation rules

(1) Stream tree:

$$\mathbf{STree} \ni st ::= s \mid (st_1, s)$$

(2) Convert a stream tree to a list of stream ids:

$$\begin{aligned} &\bar{\ }: \mathbf{STree} \to \mathbf{S} \\ &\overline{s} = [s] \\ &\overline{(st, s)} = \overline{st} +\!\!+ [s] \end{aligned}$$

(3) Translation environment:

$$\delta = [x_1 \mapsto st_1, ..., x_i \mapsto st_i]$$

- Judgment $\boxed{\delta \ \vdash \ e \Rightarrow_{s_1}^{s_0} (p, st)}$

$$\frac{}{\delta \ \vdash \ x \Rightarrow_{s_0}^{s_0} (\epsilon, st)} \ (\delta(x) = st) \qquad \frac{\delta \ \vdash \ e_1 \Rightarrow_{s_0'}^{s_0} (p_1, st_1) \qquad \delta[x \mapsto st_1] \ \vdash \ e_2 \Rightarrow_{s_1}^{s_0'} (p_2, st)}{\delta \ \vdash \ \textbf{let} \ x = e_1 \ \textbf{in} \ e_2 \Rightarrow_{s_1}^{s_0} (p_1; p_2, st)}$$

$$\frac{\phi(st_1, ..., st_k) \Rightarrow_{s_1}^{s_0} (p, st)}{\delta \ \vdash \ \phi(x_1, ..., x_k) \Rightarrow_{s_1}^{s_0} (p, st)} \ ((\delta(x_i) = st_i)_{i=1}^k) \qquad \left(\begin{array}{rl} \delta(y) = & (st_1, s_b) \\ (\delta(x_i) = & s_i')_{i=1}^j \\ p = & s_0 := \texttt{Usum}(s_b); \\ & (s_i := \texttt{Distr}(s_\texttt{b}, s_\texttt{i}');)_{i=1}^j \\ & \mathbf{S}_{out} := \texttt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \\ \mathbf{S}_{in} = & \texttt{fv}(p_1) \\ \mathbf{S}_{out} = & \overline{st} \cap \texttt{dv}(p_1) \\ s_{i+1} = & s_i + 1, \forall i \in \{0, ..., j-1\} \end{array} \right)$$

$$\frac{[x \mapsto st_1, (x_i \mapsto s_i)_{i=1}^j] \ \vdash \ e \Rightarrow_{s_1}^{s_0+1+j} (p_1, st)}{\delta \ \vdash \ \{e : x \ \textbf{in} \ y \ \textbf{using} \ x_1, ..., x_j\} \Rightarrow_{s_1}^{s_0} (p, (st, s_b))}$$

- Auxiliary Judgment $\boxed{\phi(st_1, ..., st_k) \Rightarrow_{s_1}^{s_0} (p, st)}$

$$\frac{}{\textbf{const}_n() \Rightarrow_{s_0+1}^{s_0} (s_0 := \texttt{Const}_\texttt{n}(), s_0)}$$

$$\frac{}{\textbf{iota}(s) \Rightarrow_{s_4}^{s_0} (p, (s_3, s_0))} \left(\begin{array}{l} s_{i+1} = s_i + 1, \forall i \in \{0, ..., 3\} \\ p = s_0 := \texttt{ToFlags}(s); \\ \quad s_1 := \texttt{Usum}(s_0); \\ \quad \overline{s_2} := \texttt{WithCtrl}(s_1, [s_1], s_2 := \texttt{Const}_\texttt{1}()); \\ \quad s_3 := \texttt{ScanPlus}_0(s_0, s_2) \end{array} \right)$$

$$\frac{}{\textbf{plus}(s_1, s_2) \Rightarrow_{s_0+1}^{s_0} (s_0 := \texttt{MapTwo}_+(s_1, s_2), s_0)}$$

3.2 Value representation

1. SVCODE values:
$$\textbf{SvVal} \ni w ::= \vec{a} \mid (w, \vec{b})$$
2. SVCODE values concatenation:
$$\begin{aligned} &++ : \textbf{SvVal} \to \textbf{SvVal} \to \textbf{SvVal} \\ &\langle \vec{a}_1, ..., \vec{a}_i \rangle ++ \langle \vec{a}_1', ..., \vec{a}_j' \rangle = \langle \vec{a}_1, ..., \vec{a}_i, \vec{a}_1', ..., \vec{a}_j' \rangle \\ &(w_1, \vec{b}_1) ++ (w_2, \vec{b}_2) = (w_1 ++ w_2, \vec{b}_1 ++ \vec{b}_2) \end{aligned}$$
3. SVCODE value construction from a stream tree:
$$\begin{aligned} &\sigma : \textbf{STree} \to \textbf{SvVal} \\ &\sigma(s) = \vec{a} \\ &\sigma((st, s)) = (\sigma(st), \sigma(s)) \end{aligned}$$
4. Value representation rules
   - Judgment $\boxed{v \rhd_\tau w}$

$$\frac{}{n \rhd_{\textbf{int}} \langle n \rangle} \qquad \frac{(v_i \rhd_\tau w_i)_{i=1}^k}{\{v_1, ..., v_k\} \rhd_{\{\tau\}} (w, \langle \texttt{F}_1, ..., \texttt{F}_k, \texttt{T} \rangle)} \ (w = w_1 ++ ... ++ w_k)$$

Lemma 3.1 (**Value translation backwards determinism**). *If* $v \triangleright_\tau w$, $v' \triangleright_\tau w$, *then* $v = v'$.

3.3 Correctness proof

Lemma 3.2. *If*

(i) $\phi : (\tau_1, ..., \tau_k) \to \tau$ *(by some derivation* $\mathcal{T}$*)*

(ii) $\phi(v_1, ..., v_k) \ \vdash \ v$ *(by* $\mathcal{E}$*)*

(iii) $\phi(st_1, ..., st_k) \Rightarrow_{s_1}^{s_0} (p, st)$ *(by* $\mathcal{C}$*)*

(iv) $(v_i \triangleright_{\tau_i} \sigma(st_i))_{i=1}^k$

(v) $\bigcup_{i=1}^k \texttt{sids}(st_i) \lessdot s_0$

then

(vi) $\langle p, \sigma \rangle \downarrow^{\langle () \rangle} \sigma'$ *(by* $\mathcal{P}$*)*

(vii) $v \triangleright_\tau \sigma'(st)$ *(by* $\mathcal{R}$*)*

(viii) $\sigma' \overset{< s_0}{=\!=} \sigma$

(ix) $s_0 \leq s_1$

(x) $\texttt{sids}(st) \lessdot s_1$

Proof. By inducntion on the syntax of ϕ.

- Case $\phi = \mathbf{const}_n$
  There is only one possibility for each of $\mathcal{T}$, $\mathcal{E}$ and $\mathcal{C}$:

$$\mathcal{T} = \frac{}{\mathbf{const}_n : () \to \mathbf{int}}$$

$$\mathcal{E} = \frac{}{\vdash \ \mathbf{const}_n() \downarrow n}$$

$$\mathcal{C} = \frac{}{\mathbf{const}_n() \Rightarrow_{s_0+1}^{s_0} (s_0 := \texttt{Const}_\texttt{n}(), s_0)}$$

  So $k = 0, \tau = \mathbf{int}, v = n, p = s_0 := \texttt{Const}_\texttt{n}()$, $s_1 = s_0 + 1$, and $st = s_0$
  By P-Xducer, P-X-Loop, P-X-Termi and P-Const, we can construct $\mathcal{P}$ as follows:

$$\mathcal{P} = \frac{\dfrac{\dfrac{}{\texttt{Const}_\texttt{n}() \Downarrow \langle n \rangle} \quad \dfrac{}{\texttt{Const}_\texttt{n}() \downarrow^{\langle \rangle} \langle \rangle}}{\texttt{Const}_\texttt{n}() \downarrow^{\langle () \rangle} \langle n \rangle}}{\langle s_0 := \texttt{Const}_\texttt{n}(), \sigma \rangle \downarrow^{\langle () \rangle} \sigma[s_0 \mapsto \langle n \rangle]}$$

  So $\sigma' = \sigma[s_0 \mapsto \langle n \rangle]$.
  Then we take $\mathcal{R} = \dfrac{}{n \triangleright_{\mathbf{int}} \sigma'(s_0)}$.
  Also clearly, $\sigma' \overset{< s_0}{=\!=} \sigma$, $s_0 \leq s_0 + 1$, $\texttt{sids}(s_0) \lessdot s_0 + 1$, and we are done.

- Case $\phi = \mathbf{plus}$
  We must have
  $$\mathcal{T} = \frac{}{\mathbf{plus} : (\mathbf{int}, \mathbf{int}) \to \mathbf{int}}$$
  $$\mathcal{E} = \frac{}{\vdash\ \mathbf{plus}(n_1, n_2) \downarrow n_3}$$
  where $n_3 = n_2 + n_1$, and
  $$\mathcal{C} = \frac{}{\mathbf{plus}(s_1, s_2) \Rightarrow^{s_0}_{s_0+1} (s_0 := \mathtt{MapTwo}_+(s_1, s_2), s_0)}$$
  So $k = 2, \tau_1 = \tau_2 = \tau = \mathbf{int}, v_1 = n_1, v_2 = n_2, v = n_3, st_1 = s_1, st_2 = s_2, st = s_0, s_1 = s_0 + 1$ and $p = s_0 := \mathtt{MapTwo}_+(s_1, s_2)$.

  Assumption (iv) gives us $\overline{n_1 \rhd_{\mathbf{int}} \sigma(s_1)}$ and $\overline{n_2 \rhd_{\mathbf{int}} \sigma(s_2)}$, which implies $\sigma(s_1) = \langle n_1 \rangle$ and $\sigma(s_2) = \langle n_2 \rangle$ respectively.

  For (v) we have $s_1 < s_0$ and $s_2 < s_0$.

  Then using P-Xducer with $\sigma(s_1) = \langle n_1 \rangle$ and $\sigma(s_2) = \langle n_2 \rangle$, and using P-X-Loop and P-X-Termi, we can build $\mathcal{P}$ as follows:
  $$\frac{\dfrac{\overline{\mathtt{MapTwo}_+(\langle n_1 \rangle, \langle n_2 \rangle) \Downarrow \langle n_3 \rangle} \quad \overline{\mathtt{MapTwo}_+(\langle\rangle, \langle\rangle) \downarrow^{\langle\rangle} \langle\rangle}}{\mathtt{MapTwo}_+(\langle n_1 \rangle, \langle n_2 \rangle) \downarrow^{\langle() \rangle} \langle n_3 \rangle}}{\left\langle s_0 := \mathtt{MapTwo}_+(s_1, s_2), \sigma \right\rangle \downarrow^{\langle() \rangle} \sigma[s_0 \mapsto \langle n_3 \rangle]}$$
  Therefore, $\sigma' = \sigma[s_0 \mapsto \langle n_3 \rangle]$.
  Now we can take $\mathcal{R} = \overline{n_3 \rhd_{\mathbf{int}} \sigma'(s_0)}$, and it is clear that $\sigma' \overset{<s_0}{=\!=} \sigma$, $s_0 \leq s_0 + 1$ and $\mathtt{sids}(s_0) \lessdot s_0 + 1$ as required.
- Case $\phi = \mathbf{iota}$

□

Theorem 3.3. ***If***

(i) $\Gamma\ \vdash\ e : \tau$ *(by some derivation* $\mathcal{T}$*)*

(ii) $\rho\ \vdash\ e \downarrow v$ *(by some* $\mathcal{E}$*)*

(iii) $\delta\ \vdash\ e \Rightarrow^{s_0}_{s_1} (p, st)$ *(by some* $\mathcal{C}$*)*

(iv) $\forall x \in dom(\Gamma).\ \vdash\ \rho(x) : \Gamma(x) \wedge \mathtt{sids}(\delta(x)) \lessdot s_0 \wedge \rho(x) \rhd_{\Gamma(x)} \sigma(\delta(x))$
then

(v) $\langle p, \sigma \rangle \downarrow^{\langle() \rangle} \sigma'$ *(by some derivation* $\mathcal{P}$*)*

(vi) $v \rhd_\tau \sigma'(st)$ *(by some* $\mathcal{R}$*)*

(vii) $\sigma' \overset{<s_0}{=\!=} \sigma$

(viii) $s_0 \leq s_1$

(ix) $\mathtt{sids}(st) \lessdot s_1$

Proof. By induction on the syntax of e.

- Case $e = \{e_1 : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\}$.

  We must have:

  (i)
  $$\mathcal{T} = \frac{\begin{matrix}\mathcal{T}_1 \\ [x \mapsto \tau_1, (x_i \mapsto \textbf{int})_{i=1}^{j}] \ \vdash \ e_1 : \tau_2\end{matrix}}{\Gamma \ \vdash \ \{e_1 : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\} : \{\tau_2\}}$$
  and
  $$\Gamma(y) = \{\tau_1\}$$
  $$(\Gamma(x_i) = \textbf{int})_{i=1}^{j}$$

  (ii)
  $$\mathcal{E} = \frac{\begin{matrix}\mathcal{E}_i \\ ([x \mapsto v_i, (x_i \mapsto n_i)_{i=1}^{j}] \ \vdash \ e_1 \downarrow v_i')_{i=1}^{k}\end{matrix}}{\rho \ \vdash \ \{e_1 : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\} \downarrow \{v_1', ..., v_k'\}}$$
  and
  $$\rho(y) = \{v_1, ..., v_k\}$$
  $$(\rho(x_i) = n_i)_{i=1}^{j}$$

  (iii)
  $$\mathcal{C} = \frac{\begin{matrix}\mathcal{C}_1 \\ [x \mapsto st_1, (x_i \mapsto s_i)_{i=1}^{j}] \ \vdash \ e_1 \Rightarrow_{s_1}^{s_0+1+j} (p_1, st_2)\end{matrix}}{\delta \ \vdash \ \{e_1 : x \textbf{ in } y \textbf{ using } x_1, ..., x_j\} \Rightarrow_{s_1}^{s_0} (p, (st_2, s_b))}$$
  and
  $$\begin{aligned}
  \delta(y) &= (st_1, s_b) \\
  (\delta(x_i) &= s_i')_{i=1}^{j} \\
  p &= s_0 := \texttt{Usum}(s_b); \\
  &\quad (s_i := \texttt{Distr}(s_\texttt{b}, s_\texttt{i}');)_{i=1}^{j} \\
  &\quad \mathbf{S}_{out} := \texttt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \\
  \mathbf{S}_{in} &= \texttt{fv}(p_1) \\
  \mathbf{S}_{out} &= \overline{st_2} \cap \texttt{dv}(p_1) \\
  s_{i+1} &= s_i + 1, \forall i \in \{0, ..., j-1\}
  \end{aligned}$$

  So $\tau = \{\tau_2\}, v = \{v_1', ..., v_k'\}, st = (st_2, s_b)$.

  (iv) $\vdash \ \rho(y) : \Gamma(y)$ gives us $\vdash \ \{v_1, ..., v_k\} : \{\tau_1\}$, which must have the derivation:
  $$\frac{(\vdash \ v_i : \tau_1)_{i=1}^{k}}{\vdash \ \{v_1, ..., v_k\} : \{\tau_1\}} \tag{3.1}$$

  $\overline{\delta(y)} \lessdot s_0$ gives us
  $$\overline{\delta(y)} = \overline{(st_1, s_b)} = \overline{st_1} +\!\!+ [s_b] \lessdot s_0 \tag{3.2}$$

and $(\delta(x_i) = s'_i)_{i=1}^{j} \lessdot s_0$ implies and

$$[s'_1, ..., s'_j] \lessdot s_0 \tag{3.3}$$

Since $\rho(y) \triangleright_{\Gamma(y)} \sigma(\delta(y)) = \{v_1, ..., v_k\} \triangleright_{\{\tau_1\}} \sigma((st_1, s_b))$, which must have the derivation:

$$\frac{\begin{array}{c}\mathcal{R}_i \\ (v_i \triangleright_{\tau_1} w_i)_{i=1}^{k}\end{array}}{\{v_1, ..., v_k\} \triangleright_{\{\tau_1\}} (w, \langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle)} (w = w_1 \mathbin{+\!\!+} ... \mathbin{+\!\!+} w_k) \tag{3.4}$$

therefore

$$\sigma(st_1) = w \tag{3.5}$$

and

$$\sigma(s_b) = \langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle. \tag{3.6}$$

First we shall show:

(v) $\left\langle \begin{array}{l} s_0 := \mathtt{Usum}(s_b); \\ (s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i});)_{i=1}^{j} \\ \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \end{array} , \sigma \right\rangle \downarrow^{\langle () \rangle} \sigma'$ by some $\mathcal{P}$

(vi) $\{v'_1, ..., v'_k\} \triangleright_{\{\tau_2\}} \sigma'((st_2, s_b))$ by some $\mathcal{R}$

(vii) $\sigma' \overset{<s_0}{=\!=} \sigma$

Using P-Seq $(j+1)$ times, we can build $\mathcal{P}$ as follows:

$$\dfrac{\mathcal{P}_0 \quad \langle s_0 := \mathtt{Usum}(s_b), \sigma \rangle \downarrow^{\langle () \rangle} \sigma_0 \qquad \dfrac{\mathcal{P}_1 \quad \langle s_1 := \mathtt{Distr}(s_\mathtt{b}, s'_1), \sigma_0 \rangle \downarrow^{\langle () \rangle} \sigma_1 \qquad \dfrac{\mathcal{P}_{j+1} \quad \dfrac{\mathcal{P}_j \quad \langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1), \sigma_j \rangle \downarrow^{\langle () \rangle} \sigma'}{\vdots} \quad \left\langle \begin{array}{l} (s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i});)_{i=1}^{j} \\ \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \end{array} , \sigma_1 \right\rangle \downarrow^{\langle () \rangle} \sigma'}{\left\langle \begin{array}{l} (s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i});)_{i=1}^{j} \\ \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \end{array} , \sigma_1 \right\rangle \downarrow^{\langle () \rangle} \sigma'}}{\left\langle \begin{array}{l} s_0 := \mathtt{Usum}(s_b); \\ (s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i});)_{i=1}^{j} \\ \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1) \end{array} , \sigma \right\rangle \downarrow^{\langle () \rangle} \sigma'}$$

in which for $\forall i \in \{1, ..., j\}$, $\mathcal{P}_i$ is derivation of $\langle s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i}), \sigma_{i-1} \rangle \downarrow^{\langle () \rangle} \sigma_i$. For $\mathcal{P}_0$, with $\sigma(s_b) = \langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle$, we can build it as follows:

$$\text{by P-X-Loop} \dfrac{\text{by P-Usum}\mathtt{F} \dfrac{\text{by P-Usum}\mathtt{T} \dfrac{}{\mathtt{Usum}(\langle \mathtt{T} \rangle) \Downarrow \langle \rangle}}{\vdots} \quad \mathtt{Usum}(\langle \mathtt{F}_2, ..., \mathtt{F}_k, \mathtt{T} \rangle) \Downarrow \langle ()_2, ..., ()_k \rangle}{\mathtt{Usum}(\langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle) \Downarrow \langle ()_1, ..., ()_k \rangle} \quad \text{by P-X-Termi} \dfrac{}{\mathtt{Usum}(\langle \rangle) \downarrow^{\langle \rangle} \langle \rangle}$$

$$\text{by P-Xducer} \dfrac{\mathtt{Usum}(\langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle) \downarrow^{\langle () \rangle} \vec{a}_0}{\langle s_0 := \mathtt{Usum}(s_b), \sigma \rangle \downarrow^{\langle () \rangle} \sigma[s_0 \mapsto \vec{a}_0]}$$

So $\sigma_0 = \sigma[s_0 \mapsto \vec{a}]$, and $\vec{a}_0 = \langle ()_1, ..., ()_k \rangle$.

Similarly, with $\sigma(s_b) = \langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle$ and $\sigma(s'_i) = \langle n_i \rangle$, we can build each $\mathcal{P}_i$ for $\forall i \in \{1, ..., j\}$ as follows:

$$\text{by P-DistrT } \frac{}{\mathtt{Distr}(\langle \mathtt{T} \rangle, \langle n_i \rangle) \Downarrow \langle\rangle}$$

$$\vdots$$

$$\text{by P-DistrF } \frac{\mathtt{Distr}(\langle \mathtt{F}_2, ..., \mathtt{F}_k, \mathtt{T} \rangle, \langle n_i \rangle) \Downarrow \langle \overbrace{n_i, ..., n_i}^{k-1} \rangle}{}$$

$$\text{by P-X-Loop } \frac{\mathtt{Distr}(\langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle, \langle n_i \rangle) \Downarrow \langle \overbrace{n_i, ..., n_i}^{k} \rangle}{} \qquad \text{by P-X-TermI } \frac{}{\mathtt{Distr}(\langle\rangle, \langle\rangle) \downarrow^{\langle\rangle} \langle\rangle}$$

$$\text{by P-Xducer } \frac{\mathtt{Distr}(\langle \mathtt{F}_1, ..., \mathtt{F}_k, \mathtt{T} \rangle, \langle n_i \rangle) \downarrow^{\langle()\rangle} \vec{a}_i}{\langle s_i := \mathtt{Distr}(s_\mathtt{b}, s'_\mathtt{i}), \sigma_{i-1} \rangle \downarrow^{\langle()\rangle} \sigma_{i-1}[s_i \mapsto \vec{a}_i]}$$

So $\forall i \in \{1, ..., j\}.\sigma_i = \sigma_{i-1}[s_i \mapsto \vec{a}_i]$ and $\vec{a}_i = \langle \overbrace{n_i, ..., n_i}^{k} \rangle$.

Thus $\sigma_j = \sigma[s_0 \mapsto \langle()_1, ..., ()_k\rangle, s_1 \mapsto \langle \overbrace{n_1, ..., n_1}^{k} \rangle, ..., s_j \mapsto \langle \overbrace{n_j, ..., n_j}^{k} \rangle]$.

Now it remains to build $\mathcal{P}_{j+1}$.

Since we have

$$\mathcal{T}_1 = \ [x \mapsto \tau_1, (x_i \mapsto \mathbf{int})_{i=1}^{j}] \ \vdash \ e_1 : \tau_2$$

$$(\mathcal{E}_i = \ [x \mapsto v_i, (x_i \mapsto n_i)_{i=1}^{j}] \ \vdash \ e_1 \downarrow v'_i \qquad)_{i=1}^{k}$$

$$\mathcal{C}_1 = \ [x \mapsto st_1, (x_i \mapsto s_i)_{i=1}^{j}] \ \vdash \ e_1 \Rightarrow_{s_1}^{s_0+1+j} (p_1, st_2)$$

Let $\Gamma_1 = [x \mapsto \tau_1, (x_i \mapsto \mathbf{int})_{i=1}^{j}], \rho_i = [x \mapsto v_i, (x_i \mapsto n_i)_{i=1}^{j}]$ and $\delta_1 = [x \mapsto st_1, (x_i \mapsto s_i)_{i=1}^{j}]$.

From (**??**) and (**??**) it is clear that

$$\forall z \in dom(\Gamma_1). \ \vdash \ \rho_i(z) : \Gamma_1(z) \wedge \overline{\delta_1(z)} \lessdot s_0 + 1 + j.$$

Let i range from 1 to k: we take $\sigma'_i \overset{\mathbf{S}}{\sim} \sigma_j$ where $\mathbf{S} = dom(\sigma_j) - \overline{st_1} - \{s_1, ..., s_j\}$, such that

$$\sigma'_i(st_1) = w_i \tag{3.7}$$

$$(\sigma'_i(s_i) = \langle n_i \rangle)_{i=1}^{j} \tag{3.8}$$

It is easy to show that

$$\sigma'_1 \overset{\mathbf{S}}{\sim} \sigma'_2 \overset{\mathbf{S}}{\sim} ... \overset{\mathbf{S}}{\sim} \sigma'_k \overset{\mathbf{S}}{\sim} \sigma_j \tag{3.9}$$

$$(\overset{\mathbf{S}}{\bowtie} \sigma'_i)_{i=1}^{k} = \sigma_j \tag{3.10}$$

Also note that we now have

$$\mathbf{S}_{in} = \mathtt{fv}(p_1) \subseteq (st_1 \cup \{s_1, ..., s_j\}) \cap \mathbf{S} = \emptyset \tag{3.11}$$

From $\mathcal{R}_i$ in (3.4) we know that

$$\forall z \in dom(\Gamma_1).\rho_i(z) \rhd_{\Gamma_1(z)} \sigma'_i(\delta_1(z)).$$

Then by IH (k times) on $\mathcal{T}_1$ with $\mathcal{E}_i$, $\mathcal{C}_1$ we obtain the following result:

$$(\langle p_1, \sigma_i' \rangle \downarrow^{\langle () \rangle} \sigma_i'')_{i=1}^k \tag{3.12}$$

$$(v_i' \triangleright_{\tau_2} \sigma_i''(st_2))_{i=1}^k \tag{3.13}$$

$$(\sigma_i'' \overset{<s_0+j+1}{=\!=\!=\!=} \sigma_i')_{i=1}^k \tag{3.14}$$

$$s_0 + 1 + j \leq s_1 \tag{3.15}$$

$$\overline{st_2} \lessdot s_1 \tag{3.16}$$

Assume $\mathbf{S}_{out} = \{s_{j+1}, ..., s_{j+l}\}$.(Note here s_{j+i} is not necessary equal to $s_j + i$, but must be $\geq s_j$).

There are two possibilities for $\mathcal{P}_{j+1}$:

– Subcase $\sigma_j(s_0) = \langle\rangle$, i.e., $k = 0$.
Then $(\sigma_j(s_i) = \langle\rangle)_{i=1}^j$. Also, with (3.4) and (3.5), we have $\forall s \in \overline{st_1}.\sigma_j(s) = \langle\rangle$; with (3.6), $\sigma_j(s_b) = \langle \mathtt{T} \rangle$. Thus

$$\forall s \in (\{s_0\} \cup \mathbf{S}_{in}).\sigma_j(s) = \langle\rangle$$

Then we can use the rule P-Wc-Emp to build $\mathcal{P}$ as follows:

$$\frac{}{\langle \mathbf{S}_{out} := \mathtt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1), \sigma_j \rangle \downarrow^{\langle () \rangle} \sigma_j[(s_{j+i} \mapsto \langle\rangle)_{i=1}^l]}$$

So in this subcase,

$$\sigma' = \sigma_j[(s_{j+i} \mapsto \langle\rangle)_{i=1}^l] = \sigma[s_0 \mapsto \langle\rangle, s_1 \mapsto \langle\rangle, ..., s_{j+l} \mapsto \langle\rangle].$$

TS: (vi)
Since $k = 0$, then $v = \{\}$. Also, we have

$$\sigma'(s_b) = \sigma(s_b) = \langle \mathtt{T} \rangle$$

$$\forall s \in \overline{st_2}.\sigma'(s) = \langle\rangle$$

Therefore, $\sigma'((st_2, s_b)) = (\sigma'(st_2), \sigma'(s_b))$, with which we construct

$$\mathcal{R} = \frac{}{\{\} \triangleright_{\{\tau_2\}} ((...(\langle\rangle, (\langle\rangle)), ...), \langle \mathtt{T} \rangle)}$$

as required.

– Subcase $\sigma_j(s_0) = \langle ()|... \rangle$, i.e., $k > 0$.
Since we have (3.9), (3.12) and $fv(p_1) \cap \mathbf{S} = \emptyset$ from (3.11), it is easy to show that using Lemma 2.6 at most (k-1) times we can obtain

$$\left\langle p_1, (\overset{\mathbf{S}}{\bowtie} \sigma_i')_{i=1}^k \right\rangle \downarrow^{\langle ()_1, ..., ()_k \rangle} (\overset{\mathbf{S}}{\bowtie} \sigma_i'')_{i=1}^k \tag{3.17}$$

Let $\sigma'' = (\overset{\mathbf{S}}{\bowtie} \sigma_i'')_{i=1}^k$. Also with (3.10), we replace both the start and ending stores in (3.17), giving us a derivation $\mathcal{P}'_{j+1}$

$$\langle p_1, \sigma_j \rangle \downarrow^{\langle ()_1, ..., ()_k \rangle} \sigma''$$

And for $\forall i \in \{1, ..., l\}$, by Definition 2.2 we have

$$\sigma''(s_{j+i}) = \sigma''_1(s_{j+i}) +\!+ ... +\!+ \sigma''_k(s_{j+i})$$

Now we can build $\mathcal{P}_{j+1}$ using the rule P-Wc-Nonemp as follows:

$$\frac{\begin{array}{c}\mathcal{P}'_{j+1} \\ \langle p_1, \sigma_j\rangle \downarrow^{\langle()_1,...,()_k\rangle} \sigma''\end{array}}{\langle \mathbf{S}_{out} := \texttt{WithCtrl}(s_0, \mathbf{S}_{in}, p_1), \sigma_j\rangle \downarrow^{\langle()\rangle} \sigma_j[(s_{j+i} \mapsto \sigma''(s_{j+i}))^l_{i=1}]}$$

So in this subcase $\sigma' = \sigma_j[(s_{j+i} \mapsto \sigma''(s_{j+i}))^l_{i=1}]$.

TS : (vi)
Let $\sigma'(st_2) = w'$, and $\sigma''_i(st_2) = w'_i$ then it is easy to show that $w' = \sigma''(st_2) = w'_1 +\!+ ... +\!+ w'_k$.
Also, $\sigma'(s_b) = \sigma(s_b) = \langle \texttt{F}_1, ..., \texttt{F}_k, \texttt{T}\rangle$, we now have $\sigma'((st_2, s_b)) = (\sigma'(st_2), \sigma'(s_b)) = (w', \langle \texttt{F}_1, ..., \texttt{F}_k, \texttt{T}\rangle)$. With (3.13), we can construct $\mathcal{R}$ as follows:

$$\frac{(v'_i \rhd_{\tau_2} w'_i)^k_{i=1}}{\{v'_1, ..., v'_k\} \rhd_{\{\tau_2\}} (w', \langle \texttt{F}_1, ..., \texttt{F}_k, \texttt{T}\rangle)}$$

as required.

(viii) TS: (vii)
It is clear that in both subcase we have

$$dom(\sigma') = dom(\sigma) \cup \{s_0\} \cup \{s_1, ..., s_j\} \cup \{s_{j+1}, ..., s_{j+l}\}$$

Since $\forall s \in \{s_0\} \cup \{s_1, ..., s_j\} \cup \{s_{j+1}, ..., s_{j+l}\}. s_0 \leq s$, therefore $\sigma' \overset{<s_0}{=\!=} \sigma$.

(ix) TS: $s_0 \leq s_1$
From (3.15) we immediately get $s_0 \leq s_1 - 1 - j < s_1$.

(x) TS: $\overline{(st_2, s_b)} \lessdot s_1$
From (**??**) we know $s_b < s_0$, thus $s_b < s_0 \leq s_1$. And we already have (3.16). Therefore,

$$\overline{(st_2, s_b)} = \overline{st_2} +\!+ \{s_b\} \lessdot s_1.$$

- Case $e = x$.
We must have

$$\mathcal{T} = \frac{}{\Gamma \vdash x : \tau}\,(\Gamma(x) = \tau)$$
$$\mathcal{E} = \frac{}{\rho \vdash x \downarrow v}\,(\rho(x) = v)$$
$$\mathcal{C} = \frac{}{\delta \vdash x \Rightarrow^{s_0}_{s_0} (\epsilon, st)}\,(\delta(x) = st)$$

So $p = \epsilon$.
Immediately we have $\mathcal{P} = \dfrac{}{\langle \epsilon, \sigma\rangle \downarrow^{\langle()\rangle} \sigma}$
So $\sigma' = \sigma$, which implies $\sigma' \overset{<s_0}{=\!=} \sigma$.
From the assumption we already have $v \rhd_\tau \sigma(st)$, and $\overline{st} \lessdot s_0$.
Finally it's clear that $s_0 \leq s_0$, and we are done.

- Case $e = \textbf{let}\ x = e_1\ \textbf{in}\ e_2$.

  We must have:

$$\mathcal{T} = \frac{\begin{array}{cc}\mathcal{T}_1 & \mathcal{T}_2\end{array}}{\begin{array}{cc}\Gamma \vdash e_1 : \tau_1 & \Gamma[x \mapsto \tau_1] \vdash e_2 : \tau\end{array}}{\Gamma \vdash \textbf{let}\ x = e_1\ \textbf{in}\ e_2 : \tau}$$

$$\mathcal{E} = \frac{\begin{array}{cc}\mathcal{E}_1 & \mathcal{E}_2\end{array}}{\begin{array}{cc}\rho \vdash e_1 \downarrow v_1 & \rho[x \mapsto v_1] \vdash e_2 \downarrow v\end{array}}{\rho \vdash \textbf{let}\ x = e_1\ \textbf{in}\ e_2 \downarrow v}$$

$$\mathcal{C} = \frac{\begin{array}{cc}\mathcal{C}_1 & \mathcal{C}_2\end{array}}{\begin{array}{cc}\delta \vdash e_1 \Rightarrow_{s_0'}^{s_0} (p_1, st_1) & \delta[x \mapsto st_1] \vdash e_2 \Rightarrow_{s_1}^{s_0'} (p_2, st)\end{array}}{\delta \vdash \textbf{let}\ x = e_1\ \textbf{in}\ e_2 \Rightarrow_{s_1}^{s_0} (p_1; p_2, st)}$$

  So $p = p_1; p_2$.

  By IH on $\mathcal{T}_1$ with $\mathcal{E}_1, \mathcal{C}_1$, we get

  (a) $\mathcal{P}_1$ of $\langle p_1, \sigma\rangle \downarrow^{\langle()\rangle} \sigma_1$
  (b) $\mathcal{R}_1$ of $v_1 \triangleright_{\tau_1} \sigma_1(st_1)$
  (c) $\sigma_1 \overset{<s_0}{=\!=} \sigma$
  (d) $s_0 \leq s_0'$
  (e) $\overline{st_1} \lessdot s_0'$

  From (b), we know $\rho[x \mapsto v_1](x) : \Gamma[x \mapsto \tau_1](x)$ and $\rho[x \mapsto v_1](x) \triangleright_{\Gamma[x \mapsto \tau_1](x)} \sigma_1(\delta[x \mapsto st_1](x))$ must hold. From (e), we have $\overline{\delta[x \mapsto st_1](x)} \lessdot s_0'$.
  Then by IH on $\mathcal{T}_2$ with $\mathcal{E}_2, \mathcal{C}_2$, we get

  (f) $\mathcal{P}_2$ of $\langle p_2, \sigma_1\rangle \downarrow^{\langle()\rangle} \sigma_2$
  (g) $\mathcal{R}_2$ of $\sigma_2 \triangleright_{\tau} \sigma_2(st)$
  (h) $\sigma_2 \overset{<s_0'}{=\!=} \sigma_1$
  (i) $s_0' \leq s_1$
  (j) $\overline{st} \lessdot s_1$

  So we can construct:

$$\mathcal{P} = \frac{\begin{array}{cc}\mathcal{P}_1 & \mathcal{P}_2\end{array}}{\begin{array}{cc}\langle p_1, \sigma\rangle \downarrow^{\langle()\rangle} \sigma_1 & \langle p_2, \sigma_1\rangle \downarrow^{\langle()\rangle} \sigma_2\end{array}}{\langle p_1; p_2, \sigma\rangle \downarrow^{\langle()\rangle} \sigma_2}$$

  From (c), (d) and (h), it is clear that $\sigma_2 \overset{<s_0}{=\!=} \sigma_1 \overset{<s_0}{=\!=} \sigma$. From (d) and (i), $s_0 \leq s_1$. Take $\sigma' = \sigma_2$ (thus $\mathcal{R} = \mathcal{R}_2$) and we are done.

- Case $e = \phi(x_1, ..., x_k)$
  We must have

$$\mathcal{T} = \frac{\begin{matrix}\mathcal{T}_1 \\ \phi : (\tau_1, ..., \tau_k) \to \tau\end{matrix}}{\Gamma \ \vdash \ \phi(x_1, ..., x_k) : \tau} ((\Gamma(x_i) = \tau_i)_{i=1}^k)$$

$$\mathcal{E} = \frac{\begin{matrix}\mathcal{E}_1 \\ \vdash \ (v_1, ..., v_k)(\downarrow)\end{matrix} \ \vdash \ v}{\rho \ \vdash \ \phi(x_1, ..., x_k) \downarrow v} ((\rho(x_i) = v_i)_{i=1}^k)$$

$$\mathcal{C} = \frac{\begin{matrix}\mathcal{C}_1 \\ \phi(st_1, ..., st_k) \Rightarrow_{s_1}^{s_0} (p, st)\end{matrix}}{\delta \ \vdash \ \phi(x_1, ..., x_k) \Rightarrow_{s_1}^{s_0} (p, st)} ((\delta(x_i) = st_i)_{i=1}^k)$$

  From our assumption (iv), for all $i \in \{1, ..., k\}$:

  (a) $\vdash \ \rho(x_i) : \Gamma(x_i)$, that is, $\vdash \ v_i : \tau_i$

  (b) $\overline{\delta(x_i)} \lessdot s_0$, that is, $\overline{st_i} \lessdot s_0$

  (c) $\rho(x_i) \rhd_{\Gamma(x_i)} \sigma(st_i)$, that is, $v_i \rhd_{\tau_i} \sigma(st_i)$

  So using Lemma 3.2 on $\mathcal{T}_1, \mathcal{E}_1, \mathcal{C}_1, (a), (b)$ and (c) gives us exactly what we shall show.

□