CS611 Final Solution Set

December 14, 1998

1

We add the command $c_1 \ddagger c_2$ to IMP in addition to adding break. The configuration $\langle \text{break}, \sigma \rangle$ is now terminal and corresponds to an uncaught break.
The small-step reduction rules for commands are:

$$\langle \text{skip}; c_2, \sigma \rangle \mapsto_1 \langle c_2, \sigma \rangle$$

$$\langle \text{break}; c_2, \sigma \rangle \mapsto_1 \langle \text{break}, \sigma \rangle$$

$$\frac{\langle c_1, \sigma \rangle \mapsto_1 \langle c_1', \sigma' \rangle}{\langle c_1; c_2, \sigma \rangle \mapsto_1 \langle c_1'; c_2, \sigma' \rangle}$$

$$\langle \text{skip} \ddagger c_2, \sigma \rangle \mapsto_1 \langle c_2, \sigma \rangle$$

$$\langle \text{break} \ddagger c_2, \sigma \rangle \mapsto_1 \langle \text{skip}, \sigma \rangle$$

$$\frac{\langle c_1, \sigma \rangle \mapsto_1 \langle c_1', \sigma' \rangle}{\langle c_1 \ddagger c_2, \sigma \rangle \mapsto_1 \langle c_1' \ddagger c_2, \sigma' \rangle}$$

$$\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto_1 \langle \text{if } b \text{ then } (c \ddagger \text{while } b \text{ do } c \text{ else skip}), \sigma \rangle$$

2

(a)
$v_0 = \text{nil}_{\text{unit}}$
$v_{n+1} = \text{cons}(\langle\rangle, v_n)$

(b)
let $b = \lambda x{:}\text{unit}.\lambda y{:}\text{list[unit]}.\text{cons}(\langle\rangle, y)$
$v_+ = \lambda e_1{:}\text{list[unit]}.\lambda e_2{:}\text{list[unit]}.\text{fold}(b, e_1, e_2)$

Lemma 2.1. *If* $e \mapsto^* v_m$ *then* $\text{fold}(b, e, v_n) \mapsto^* v_{n+m}$

Proof:
We proceed by induction on n.

- Base Case : $n = 0$

$$\text{fold}(b, e, v_0) = \text{fold}(b, e, \text{nil}_{\text{unit}}) \mapsto e \mapsto^* v_m = v_{m+1}$$

- Inductive Case : $n + 1$

$$\text{fold}(b, e, v_{n+1}) = \text{fold}(b, e, \text{cons}(\langle\rangle, v_n)) \mapsto \text{fold}(b, b\langle\rangle e, v_n)$$

But $b\langle\rangle e \mapsto^* \text{cons}(\langle\rangle, e) \mapsto^* \text{cons}(\langle\rangle, v_m) = v_{m+1}$. Thus by the inductive hypothesis:

$$\text{fold}(b, b\langle\rangle e, v_n) \mapsto^* v_{m+1+n} = v_{m+n+1}$$

$\square$

Theorem 2.2. $v_+ v_m v_n \mapsto^* v_{m+n}$

Proof:
$v_+ v_m v_n \mapsto^* \text{fold}(b, v_m, v_n) \mapsto^* v_{m+n}$ by the lemma. $\square$
(c)
$v_f = 0$
$v_t = 1$
$v_{if\_then\_else} = \lambda e_1{:}\text{list[unit]}.\lambda e_2{:}\tau.\lambda e_3{:}\tau.\text{fold}(\lambda x{:}\text{unit}.\lambda y{:}\tau.e_2, e_3, e_1)$
(d)

$$\frac{e_3 \mapsto e_3'}{\text{fold}(e_1, e_2, e_3) \mapsto \text{fold}(e_1, e_2, e_3')}$$

$$\text{fold}(e_1, e_2, \text{nil}_\tau) \mapsto e_2$$

$$\frac{e_1 \mapsto e_1'}{\text{fold}(e_1, e_2, \text{cons}(v_3, v_4)) \mapsto \text{fold}(e_1', e_2, \text{cons}(v_3, v_4))}$$

$$\text{fold}(v_1, e_2, \text{cons}(v_3, v_4)) \mapsto \text{fold}(v_1, (v_1 v_3 v_2), v_4)$$

(e) This problem was tricky because

$$\mathcal{T}_{\text{list}[t]} = \left\{ e \;\middle|\; \begin{array}{l} e : \text{list}[t] \text{ and } e \text{ evaluates to nil}_t \text{ or } e \text{ evaluates to} \\ \text{cons}(v_1, v_2) \text{ and } \; v_1 \in \mathcal{T}_t \text{ and } v_2 \in T_{\text{list}[t]} \end{array} \right\}$$

is *not* a definition. Instead we need to define this with

$\mathcal{T}^0_{\text{list}[\tau]} = \{e \mid e : \text{list}[\tau] \text{ and e evaluates to nil}_\tau\}$
$\mathcal{T}^{n+1}_{\text{list}[\tau]} = \{e \mid e : \text{list}[\tau] \text{ and } e \text{ evaluates to cons}(v_1, v_2) \text{ and } v_1 \in \mathcal{T}_\tau \text{ and } v_2 \in \mathcal{T}^n_{\text{list}[\tau]}\ \}$
$\mathcal{T}_{\text{list}[\tau]} = \bigcup_{i \geq 0} \mathcal{T}^i_{\text{list}[\tau]}$.

We can extend the proof in class that every $\lambda^{\rightarrow}$ term terminates, by extending the following lemma to handle term in λ^{list}.

Lemma 2.3. *If $\Gamma \vdash e : \tau$ then $\gamma(e) \in \mathcal{T}_\tau$ for all $\gamma \in \mathcal{T}_\Gamma$.*

Proof is by induction on the derivation of $\Gamma \vdash e : \tau$.

case $\Gamma \vdash \langle\rangle : \text{unit}$.

Pick $\gamma \in \mathcal{T}_\Gamma$. $\gamma(\langle\rangle) = \langle\rangle$ which is already a value, so $\gamma(\langle\rangle) \in \mathcal{T}_{\text{unit}}$.

case variable, application, abstraction

Same as in class.

case $\Gamma \vdash \text{nil}_\tau : \text{list}[\tau]$

Pick $\gamma \in \mathcal{T}_\Gamma$. $\gamma(\text{nil}_\tau) = \text{nil}_\tau$ which is already a value, so $\gamma(\text{nil}_\tau) \in \mathcal{T}_{\text{list}[\tau]}$.

case Cons

$$\frac{\Gamma \vdash e_1 : \tau \qquad \Gamma \vdash e_2 : \text{list}[\tau]}{\text{cons}(e_1, e_2) : \text{list}[\tau]}$$

Follows from the inductive hypothesis in a straightforward manner.

case Fold

$$\frac{\Gamma \vdash e_1 : \tau \to \tau_2 \to \tau_2 \qquad \Gamma \vdash e_2 : \tau_2 \qquad \Gamma \vdash e_3 : \text{list}[\tau]}{\text{fold}(e_1, e_2, e_3) : \tau_2}$$

Pick $\gamma \in \mathcal{T}_\Gamma$. The substitution distributes over the fold, so $\gamma(\text{fold}(e_1, e_2, e_2) = \text{fold}(\gamma(e_1), \gamma(e_2), \gamma(e_3))$. We know by induction that $\gamma(e_1) \in \mathcal{T}_{\tau \to \tau_2 \to \tau_2}$, $\gamma(e_2) \in \mathcal{T}_{\tau_2}$, and $\gamma(e_3) \in \mathcal{T}_{\text{list}[\tau]}$.

We need to show that for all $i \geq 0$, if e'' is in $\mathcal{T}^i_{\text{list}[\tau]}$, then for all e' in $\mathcal{T}_{\tau_2}$, $\text{fold}(\gamma(e_1), e', e'')$ is in $\mathcal{T}_{\tau_2}$.

We do this by induction on i. Case $i = 0$, if e'' is in $\mathcal{T}^0_{\text{list}[\tau]}$, then e'' evaluates to nil_τ, and $\text{fold}(\gamma(e_1), e', e'')$ evaluates to e''. For the inductive step, suppose e'' is in $\mathcal{T}^{i+1}_{\text{list}[\tau]}$, and e' is in $\mathcal{T}_{\tau_2}$. Then e'' evaluates to $\text{cons}(v_1, v_2)$ where $v_1 \in \mathcal{T}_\tau$ and $v_2 \in \mathcal{T}^i_{\text{list}[\tau]}$. $\text{fold}(\gamma(e_1), e', e'') \mapsto \text{fold}(\gamma(e_1), (\gamma(e_1) v_1 e'), v_2)$. By the outer inductive hypothesis, $(\gamma(e_1) v_1 e')$ is in $\mathcal{T}_{\tau_2}$. So by the inner inductive hypothesis, $\text{fold}(\gamma(e_1), (\gamma(e_1) v_1 e'), v_2)$ is in $\mathcal{T}_{\tau_2}$.

Using this result, since $\gamma(e_3)$ is in $T_{\text{list}[\tau]}$, and there exists some i such that $\gamma(e_3)$ is in $T^i_{\text{list}[\tau]}$, then $\text{fold}(\gamma(e_1), \gamma(e_2), \gamma(e_3))$ is in $\mathcal{T}_{\tau_2}$.

3

(a)

Types

$\mathcal{T}[\![\, \texttt{int} \,]\!] = \mathbb{Z}_\perp$

$\mathcal{T}[\![\, \text{unit} \,]\!] = \{1\}$

$\mathcal{T}[\![\, \tau_1 \times \tau_2 \,]\!] = \mathcal{T}[\![\, \tau_1 \,]\!] \times \mathcal{T}[\![\, \tau_2 \,]\!]$

$\mathcal{T}[\![\, \text{void} \,]\!] = \{\perp_{\text{void}}\}$

$\mathcal{T}[\![\, \tau_1 + \tau_2 \,]\!] = \{\langle 1, x\rangle \mid x \in \mathcal{T}[\![\, \tau_1 \,]\!]\} \cup \{\langle 2, x\rangle \mid x \in \mathcal{T}[\![\, \tau_2 \,]\!]\} \cup \perp_{\tau_1 + \tau_2}$

$\mathcal{T}[\![\tau_1 \to \tau_2]\!] = \left[\mathcal{T}[\![\tau_1]\!] \to \mathcal{T}[\![\tau_2]\!]\right].$

Environments
$\mathcal{T}[\![\Gamma]\!] = \{\gamma \mid \mathrm{Dom}(\Gamma) = \mathrm{Dom}(\gamma) \wedge \forall x \in \mathrm{Dom}(\Gamma)(\gamma(x) \in \mathcal{T}[\![\Gamma(x)]\!])\}$

Now define the denotation of an expression e, $\mathcal{E}[\![\Gamma \vdash e : \tau]\!]$ as:

$\mathcal{E}[\![\Gamma \vdash i : \mathtt{int}]\!]\rho := i \in \mathcal{T}[\![\mathtt{int}]\!] = \mathbb{Z} \cup \{\bot_{\mathtt{int}}\}$

$\mathcal{E}[\![\Gamma \vdash x : \Gamma(x)]\!]\rho := \rho(x)$

$$\mathcal{E}[\![\Gamma \vdash e_1 \oplus e_2 : \mathtt{int}]\!]\rho := \begin{cases} \bot_{\mathtt{int}} & \text{if } \mathcal{E}[\![\Gamma \vdash e_1 : \mathtt{int}]\!]\rho = \bot_{\mathtt{int}} \\ & \text{or } \mathcal{E}[\![\Gamma \vdash e_2 : \mathtt{int}]\!]\rho = \bot_{\mathtt{int}}, \\ i_1 \oplus i_2 & \text{if } \mathcal{E}[\![\Gamma \vdash e_1 : \mathtt{int}]\!]\rho = i_1 \\ & \text{and } \mathcal{E}[\![\Gamma \vdash e_2 : \mathtt{int}]\!]\rho = i_2 \text{:} \end{cases}$$

$\mathcal{E}[\![\Gamma \vdash \langle\rangle : \mathrm{unit}]\!] = 1$

$$\mathcal{E}[\![\Gamma \vdash \langle e_1, e_2\rangle : \tau_1 \times \tau_2]\!]\rho := \begin{array}{ll} \langle d_1, d_2\rangle & \text{if } \mathcal{E}[\![\Gamma \vdash e_1 : \tau_1]\!]\rho = d_1 \in \mathcal{T}[\![\tau_1]\!] \\ & \text{and } \mathcal{E}[\![\Gamma \vdash e_2 : \tau_2]\!]\rho = d_2 \in \mathcal{T}[\![\tau_2]\!] \end{array}$$

$$\mathcal{E}[\![\Gamma \vdash \pi_i(e) : \tau_i]\!]\rho := \begin{cases} \bot_{\tau_i} & \text{if } \mathcal{E}[\![\Gamma \vdash e : \tau_1 \times \tau_2]\!]\rho = \bot_{\tau_1 \times \tau_2}, \\ d_i & \text{if } \mathcal{E}[\![\Gamma \vdash e : \tau_1 \times \tau_2]\!]\rho = \langle d_1, d_2\rangle \in \mathcal{T}[\![\tau_1 \times \tau_2]\!] \end{cases}$$

$\mathcal{E}[\![\Gamma \vdash \mathtt{in}_i^{\tau_1+\tau_2}(e_i) : \tau_1 + \tau_2]\!]\rho := \langle i, d_i\rangle$ if $\mathcal{E}[\![\Gamma \vdash e_i : \tau_i]\!]\rho = d_i \in \mathcal{T}[\![\tau_i]\!]$

$\mathcal{E}[\![\Gamma \vdash (\mathtt{case}\ e\ \mathtt{of}\ \mathtt{in}_1^{\tau_1+\tau_2}(x_1) \Rightarrow e_1 \mid \mathtt{in}_2^{\tau_1+\tau_2}(x_2) \Rightarrow e_2) : \tau]\!] :=$

$$\begin{cases} \bot_\tau & \text{if } \mathcal{E}[\![\Gamma \vdash e : \tau_1 + \tau_2]\!]\rho = \bot_{\tau_1+\tau_2}, \\ \mathcal{E}[\![\Gamma, x_1 : \tau_1 \vdash e_1 : \tau]\!]\,\rho[x_1 \mapsto d_1] & \text{if } \mathcal{E}[\![\Gamma \vdash e : \tau_1 + \tau_2]\!]\rho = \langle 1, d_1\rangle, \\ \mathcal{E}[\![\Gamma, x_2 : \tau_2 \vdash e_2 : \tau]\!]\,\rho[x_2 \mapsto d_2] & \text{if } \mathcal{E}[\![\Gamma \vdash e : \tau_1 + \tau_2]\!]\rho = \langle 2, d_2\rangle \end{cases}$$

$\mathcal{E}[\![\Gamma \vdash \lambda x : \tau_1 \,.\, e : \tau_1 \to \tau_2]\!]\rho := f \in \mathcal{T}[\![\tau_2]\!]^{\mathcal{T}[\![\tau_1]\!]}$
such that $f(d) = \mathcal{E}[\![\Gamma, x : \tau_1 \vdash e : \tau_2]\!]\,\rho[x \mapsto d]$

$\mathcal{E}[\![\Gamma \vdash e_1 e_2 : \tau]\!] := \left(\mathcal{E}[\![\Gamma \vdash e_1 : \tau' \to \tau]\!]\rho\right)\left(\mathcal{E}[\![\Gamma \vdash e_2 : \tau']\!]\rho\right)$

$\mathcal{E}[\![\Gamma \vdash \mathrm{fix}(\lambda x : \tau_1 \to \tau_2 \,.\, e) : \tau_1 \to \tau_2]\!]\rho := \bigsqcup_{i \geqslant 0} f^i(f_\bot)$,
where $f = \mathcal{E}[\![\Gamma \vdash \lambda x : \tau_1 \to \tau_2 \,.\, e : (\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]\rho$
$\in \left[\mathcal{T}[\![\tau_1 \to \tau_2]\!] \to \mathcal{T}[\![\tau_1 \to \tau_2]\!]\right]$, $\quad f^i(x) = \underbrace{f(f(\ldots f(}_{i \text{ times}} x)\ldots))$.

(b)

1. $\tau_{state} \equiv \mathrm{int} \to \mathrm{int}$
   We need to restrict ourselves to only those states with a finite number of nonzero values in their ranges. For such a state we can easily find a $\mathcal{S}[\![\sigma]\!] \in \tau_{state}$, such that $\forall i \in \mathbb{Z}\ (\mathcal{S}[\![\sigma]\!]\, i \Downarrow j) \Leftrightarrow \exists X(\mathrm{locindex}(X) = i \wedge \sigma(X) = j)$

2. $\tau_a \equiv \tau_{state} \to \mathrm{int}$
   $\mathcal{A}[\![n]\!] = \lambda s{:}\tau_{state}.n$
   $\mathcal{A}[\![X]\!] = \lambda s{:}\tau_{state}.s(\mathrm{locindex}(X))$
   $\mathcal{A}[\![a_0\ p\ a_1]\!] = \lambda s{:}\tau_{state}.p(\mathcal{A}[\![a_0]\!], \mathcal{A}[\![a_1]\!])$ (for $p \in \{+, -, *\}$)

3. $\tau_b \equiv \tau_{state} \rightarrow \text{unit} + \text{unit}$
   $\mathcal{B}[\![\,\text{true}\,]\!] = \lambda s{:}\tau_{state}.\text{in}_1^{\text{unit+unit}}\langle\rangle$
   $\mathcal{B}[\![\,\text{false}\,]\!] = \lambda s{:}\tau_{state}.\text{in}_2^{\text{unit+unit}}\langle\rangle$
   $\mathcal{B}[\![\,a_0 \leq a_1\,]\!] = \lambda s{:}\tau_{state}. \leq (\mathcal{A}[\![\,a_0\,]\!], \mathcal{A}[\![\,a_1\,]\!])$
   $\mathcal{B}[\![\,\neg B\,]\!] = \lambda s{:}\tau_{state}.\text{case } \mathcal{B}[\![\,b\,]\!]\, s \text{ of in}_1(x_1) \Rightarrow \text{in}_2^{\text{unit+unit}}\langle\rangle \mid \text{in}_2(x_2) \Rightarrow \text{in}_1^{\text{unit+unit}}\langle\rangle$
   $\mathcal{B}[\![\,b_1 \wedge b_2\,]\!] = \lambda s{:}\tau_{state}.\text{case } \mathcal{B}[\![\,b_1\,]\!]\, s \text{ of in}_1(x_1) \Rightarrow \mathcal{B}[\![\,b_2\,]\!]\, s \mid \text{in}_2(x_2) \Rightarrow \text{in}_2^{\text{unit+unit}}\langle\rangle$
   $\mathcal{B}[\![\,b_1 \vee b_2\,]\!] = \lambda s{:}\tau_{state}.\text{case } \mathcal{B}[\![\,b_1\,]\!]\, s \text{ of in}_1(x_1) \Rightarrow \text{in}_1^{\text{unit+unit}}\langle\rangle \mid \text{in}_2(x_2) \Rightarrow \mathcal{B}[\![\,b_2\,]\!]\, s$
   $\mathcal{B}[\![\,a_0 = a_1\,]\!] = \mathcal{B}[\![\,(a_0 \leq a_1) \wedge (a_1 \leq a_0)\,]\!]$

4. $\tau_c \equiv \tau_{state} \rightarrow \tau_{state}$
   $\mathcal{C}[\![\,\text{skip}\,]\!] = \lambda s{:}\tau_{state}.s$
   $$\begin{aligned}\mathcal{C}[\![\,X := a\,]\!] = \lambda s{:}\tau_{state}.\lambda n{:}\text{int}.\\ &\text{case } \leq (i,n) \text{ of}\\ &\quad \text{in}_1 \Rightarrow \text{case } \leq (n,i) \text{ of}\\ &\qquad \text{in}_1 \Rightarrow \mathcal{A}[\![\,a\,]\!]\, s\\ &\qquad \text{in}_2 \Rightarrow s(n)\\ &\text{in}_2 \Rightarrow s(n)\end{aligned}$$
   (where $i = \text{locindex}(X)$)
   $\mathcal{C}[\![\,c_0; c_1\,]\!] = \lambda s{:}\tau_{state}.\mathcal{C}[\![\,c_1\,]\!]\,(\mathcal{C}[\![\,c_0\,]\!]\, s)$
   $$\begin{aligned}\mathcal{C}[\![\,\text{if } b \text{ then } c_1 \text{ else } c_2\,]\!] = \lambda s{:}\tau_{state}.\\ &(\text{case } \mathcal{B}[\![\,b\,]\!]\, s \text{ of in}_1(x_1) \Rightarrow \mathcal{C}[\![\,c_1\,]\!]\, s \mid \text{in}_2(x_2) \Rightarrow \mathcal{C}[\![\,c_2\,]\!]\, s)\end{aligned}$$
   $$\begin{aligned}\mathcal{C}[\![\,\text{while } b \text{ do } c\,]\!] = \text{fix}(\lambda f{:}\tau_{state} \rightarrow \tau_{state}.\\ &\lambda s{:}\tau_{state}.\\ &\quad(\text{case } \mathcal{B}[\![\,b\,]\!]\, s \text{ of}\\ &\qquad \text{in}_1(x_1) \Rightarrow f(\mathcal{C}[\![\,c\,]\!]\, s)\\ &\qquad \text{in}_2(x_2) \Rightarrow s))\end{aligned}$$

(c)

Since states could have a variety of representations, we will not prove that if $\langle c, \sigma_0\rangle \mapsto \sigma$, then $\mathcal{C}[\![\,c\,]\!]\,\mathcal{S}[\![\,\sigma_0\,]\!] \Downarrow \mathcal{S}[\![\,\sigma_0\,]\!]$, but instead prove that

$$\mathcal{E}[\![\,\vdash \mathcal{C}[\![\,c\,]\!]\,\mathcal{S}[\![\,\sigma_0\,]\!] : \tau_{state}\,]\!] = \mathcal{E}[\![\,\vdash \mathcal{S}[\![\,\sigma_0\,]\!] : \tau_{state}\,]\!].$$

There are a number of valid ways to prove this fact: However, the following is probably the simplest.
Def: A value s of type τ_{state} *represents* a state σ if for all $x \in \text{Var}$, $s\,(\text{locindex}(x)) \Downarrow \sigma(x)$.

Lemma 3.1 (Aexp). *If $\langle a, \sigma\rangle \mapsto n$ then for any representation s of σ, $\mathcal{A}[\![\,a\,]\!]\, s \Downarrow n$.*

Lemma 3.2 (Bexp-True). *If $\langle b, \sigma\rangle \mapsto$ true then for any representation s of σ, $\mathcal{B}[\![\,b\,]\!]\, s \Downarrow \text{in}_1^{\text{unit+unit}}(\langle\rangle)$*

Lemma 3.3 (Bexp-False). *If $\langle b, \sigma_0\rangle \mapsto$ false then for any rpresentation s of σ $\mathcal{B}[\![\,b\,]\!]\, s \Downarrow \text{in}_2^{\text{unit+unit}}(\langle\rangle)$*

Theorem 3.4. *If $\langle c, \sigma\rangle \mapsto \sigma'$ then for any representation s of σ, $\mathcal{C}[\![\,c\,]\!]\, s \Downarrow s'$ is a representation of σ'.*

Proof: We procede by induction on the derivation of $\langle c, \sigma\rangle \mapsto \sigma'$ and let s represent σ.

case $\langle \text{skip}, \sigma\rangle \mapsto \sigma$

$\mathcal{C}[\![\, \text{skip} \,]\!]\, s \Downarrow s$, which is a representation of σ.

case Assignment

$$\frac{\langle a, \sigma\rangle \mapsto m}{\langle X := a, \sigma\rangle \mapsto \sigma[m/X]}$$

$\mathcal{C}[\![\, X := a \,]\!]\, s \Downarrow s'$ where $s' =$

$$\begin{array}{l} \lambda n{:}\text{int}. \\ \quad \text{case } \le (i,n) \text{ of} \\ \quad\quad \text{in}_1 \Rightarrow \text{case } \le (n,i) \text{ of} \\ \quad\quad\quad \text{in}_1(x_1) \Rightarrow \mathcal{A}[\![\, a \,]\!]\, s \\ \quad\quad\quad \text{in}_2(x_2) \Rightarrow s(n) \\ \quad\quad \text{in}_2 \Rightarrow s(n) \end{array}$$

where $i = \text{locindex}(X)$. The PCF term s' represents $\sigma[m/X]$ as for location X, $s(\text{locindex}(X)) \Downarrow m$, since by the lemma, $\mathcal{A}[\![\, a \,]\!]\, s \Downarrow m$, and for location $Y \neq X$, $s(\text{locindex}(Y)) \Downarrow \sigma[m/X](Y)$.

case Sequence

$$\frac{\langle c_0, \sigma\rangle \mapsto \sigma'' \qquad \langle c_1, \sigma''\rangle \mapsto \sigma'}{\langle c_0; c_1, \sigma\rangle \mapsto \sigma'}$$

By induction, $\mathcal{C}[\![\, c_0 \,]\!]\, s \Downarrow s''$ is a representation of σ''. Again by induction $\mathcal{C}[\![\, c_1 \,]\!]\, s'' \Downarrow s'$ is a representation of σ'.

But $\mathcal{C}[\![\, c_0; c_1 \,]\!]\, s = \mathcal{C}[\![\, c_1 \,]\!]\, (\mathcal{C}[\![\, c_0 \,]\!]\, s) \Downarrow s'$.

case If-true

$$\frac{\langle b, \sigma\rangle \mapsto \text{true} \qquad \langle c_1, \sigma\rangle \mapsto \sigma'}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2, \sigma\rangle \mapsto \sigma'}$$

By induction, $\mathcal{C}[\![\, c_1 \,]\!]\, s \Downarrow s'$ is a representation of σ'. By the lemma Bexp-True, $\mathcal{B}[\![\, b \,]\!]\, s \Downarrow \text{in}_1^{\text{unit+unit}}\langle\rangle$. So $\mathcal{C}[\![\, \text{if } b \text{ then } c_1 \text{ else } c_2 \,]\!]\, s =$

$$\lambda s : \tau_{state}.(\text{case } \mathcal{B}[\![\, b \,]\!]\, s \text{ of in}_1 \Rightarrow \mathcal{C}[\![\, c_1 \,]\!]\, s \mid \text{in}_2 \Rightarrow \mathcal{C}[\![\, c_2 \,]\!]\, s)s \Downarrow s'.$$

case If-false

$$\frac{\langle b, \sigma\rangle \mapsto \text{false} \qquad \langle c_2, \sigma\rangle \mapsto \sigma'}{\langle \text{if } b \text{ then } c_1 \text{ else } c_2, \sigma\rangle \mapsto \sigma'}$$

Analogous to If-true case.

case While-true

$$\frac{\langle b, \sigma\rangle \mapsto \text{true} \qquad \langle c, \sigma\rangle \mapsto \sigma'' \qquad \langle \text{while } b \text{ do } c, \sigma''\rangle \mapsto \sigma'}{\langle \text{while } b \text{ do } c, \sigma\rangle \mapsto \sigma'}$$

By induction, $\mathcal{C}[\![\, c \,]\!]\, s \Downarrow s''$ which is a representation of σ''. Also by induction, $\mathcal{C}[\![\, \text{while } b \text{ do } c \,]\!]\, s'' \Downarrow s'$ which is a representation of σ'. By the lemma Bexp-True, $\mathcal{B}[\![\, b \,]\!]\, s \Downarrow \text{in}_1^{\text{unit+unit}}\langle\rangle$. So $\mathcal{C}[\![\, \text{while } b \text{ do } c \,]\!]\, s \Downarrow s'$

case While-false

$$\frac{\langle b, \sigma \rangle \mapsto \text{false}}{\langle \text{while } b \text{ do } c, \sigma \rangle \mapsto \sigma}$$

By Bexp-False, $\mathcal{B}[\![b]\!]\, s \Downarrow \text{in}_2^{\text{unit+unit}}\langle\rangle$. So

$\mathcal{C}[\![\text{while } b \text{ do } c]\!]\, s =$

$\text{fix}(\lambda f{:}\tau_{state} \to \tau_{state}.\lambda s{:}\tau_{state}.(\text{case } \mathcal{B}[\![b]\!]\, s \text{ of in}_1(x_1) \Rightarrow f(\mathcal{C}[\![c]\!]\, s) \mid \text{in}_2(x_2) \Rightarrow s))s \Downarrow s$

□

With this theorem, and as $\lambda x{:}\text{int}.0$ represents σ_0, we may state that if $\langle c, \sigma_0 \rangle \mapsto \sigma$ then $\mathcal{C}[\![c]\!]\,(\lambda x{:}\text{int}.0) \Downarrow s$ where s represents σ. By the soundness of the denotational semantics of PCF, this means that

$$\mathcal{E}[\![\vdash \mathcal{C}[\![c]\!]\, \mathcal{S}[\![\sigma_0]\!] : \tau_{state}]\!] = \mathcal{E}[\![s]\!] = \mathcal{E}[\![\vdash \mathcal{S}[\![\sigma_0]\!] : \tau_{state}]\!]\,.$$