Logical Relations for Fine-Grained Concurrency (Expanded Technical Appendix)

Aaron Turon
Northeastern University
turon@ccs.neu.edu

Jacob Thamsborg
IT University of Copenhagen
thamsborg@itu.dk

Amal Ahmed
Northeastern University
amal@ccs.neu.edu

Lars Birkedal
IT University of Copenhagen
birkedal@itu.dk

Derek Dreyer
MPI-SWS, Germany
dreyer@mpi-sws.org

September 2012

Contents

1 Language

1.1 Syntax

$$
\begin{array}{rcl}
\tau & ::= & \mathbf{1} \mid \mathbf{B} \mid \mathbf{N} \mid \alpha \mid \tau + \tau \mid \mathbf{ref}(\overline{\tau}) \mid \mathbf{ref}_?(\overline{\tau}) \mid \mu\alpha.\tau \mid \forall\alpha.\tau \mid \tau \to \tau \\
\sigma & ::= & \mathbf{1} \mid \mathbf{B} \mid \mathbf{N} \mid \tau + \tau \mid \mathbf{ref}(\overline{\tau}) \mid \mathbf{ref}_?(\overline{\tau}) \mid \mu\alpha.\sigma \\
e & ::= & () \mid \mathbf{true} \mid \mathbf{false} \mid \mathbf{if}\ e\ \mathbf{then}\ e\ \mathbf{else}\ e \mid n \mid e + e \mid x \mid \mathbf{rec}\ f(x).e \mid e\ e \mid \Lambda.e \mid e\ \_ \\
 & \mid & \mathbf{null} \mid \mathbf{some}(e) \mid \mathbf{case}(e, \mathbf{null} \Rightarrow e, \mathbf{some}(x) \Rightarrow e) \mid \mathbf{inj}_i\ e \mid \mathbf{case}(e, \mathbf{inj}_1\ x \Rightarrow e, \mathbf{inj}_2\ y \Rightarrow e) \\
 & \mid & \mathbf{new}\ \overline{e} \mid e[i] \mid e[i]\ :=\ e \mid \mathsf{CAS}(e[i], e, e) \mid \ell \mid \mathbf{fork}\ e \\
v & ::= & \mathbf{rec}\ f(x).e \mid \Lambda.e \mid () \mid n \mid \mathbf{true} \mid \mathbf{false} \mid \ell \mid x \\
a & ::= & \mathbf{new}\ \overline{v} \mid v[i] \mid v[i]\ :=\ v \mid \mathsf{CAS}(v[i], v, v) \mid \mathbf{inj}_i\ v \\
\Gamma & ::= & \cdot \mid \Gamma, x : \tau \\
\Delta & ::= & \cdot \mid \Delta, \alpha \\
\Omega & ::= & \Delta; \Gamma
\end{array}
$$

Note: recursive types are required to be *productive*: the type variable must appear under a non-μ type constructor.

1.2 Type rules

$$\Omega \vdash () : \mathbf{1} \qquad \Omega \vdash \mathbf{true} : \mathbf{B} \qquad \Omega \vdash \mathbf{false} : \mathbf{B} \qquad \Omega \vdash n : \mathbf{N} \qquad \Omega, x : \tau \vdash x : \tau$$

$$\frac{\Omega \vdash e : \mathbf{B} \qquad \Omega \vdash e_i : \tau}{\Omega \vdash \mathbf{if}\ e\ \mathbf{then}\ e_1\ \mathbf{else}\ e_2 : \tau} \qquad \frac{\Omega, f : \tau' \to \tau, x : \tau' \vdash e : \tau}{\Omega \vdash \mathbf{rec}\ f(x).e : \tau' \to \tau} \qquad \frac{\Omega \vdash e : \tau' \to \tau \qquad \Omega \vdash e' : \tau'}{\Omega \vdash e\ e' : \tau}$$

$$\Omega \vdash \mathbf{null} : \mathbf{ref}_?(\overline{\tau}) \qquad \frac{\Omega \vdash e : \mathbf{ref}(\overline{\tau})}{\Omega \vdash \mathbf{some}(e) : \mathbf{ref}_?(\overline{\tau})}$$

$$\frac{\Omega \vdash e : \mathbf{ref}_?(\overline{\tau}) \qquad \Omega \vdash e_1 : \tau \qquad \Omega, x : \mathbf{ref}(\overline{\tau}) \vdash e_2 : \tau}{\Omega \vdash \mathbf{case}(e, \mathbf{null} \Rightarrow e_1, \mathbf{some}(x) \Rightarrow e_2) : \tau} \qquad \frac{\Omega \vdash e : \tau_i}{\Omega \vdash \mathbf{inj}_i\ e : \tau_1 + \tau_2}$$

$$\frac{\Omega \vdash e : \tau_1 + \tau_2 \qquad \Omega, x : \tau_i \vdash e_i : \tau}{\Omega \vdash \mathbf{case}(e, \mathbf{inj}_1\ x \Rightarrow e_1, \mathbf{inj}_2\ x \Rightarrow e_2) : \tau} \qquad \frac{\Omega, \alpha \vdash e : \tau}{\Omega \vdash \Lambda.e : \forall\alpha.\tau} \qquad \frac{\Omega \vdash e : \forall\alpha.\tau}{\Omega \vdash e\ \_ : \tau[\tau'/\alpha]}$$

$$\frac{\Omega \vdash e_i : \tau_i}{\Omega \vdash \mathbf{new}\ (\overline{e}) : \mathbf{ref}(\overline{\tau})} \qquad \frac{\Omega \vdash e : \mathbf{ref}(\overline{\tau})}{\Omega \vdash e[i] : \tau_i} \qquad \frac{\Omega \vdash e : \mathbf{ref}(\overline{\tau}) \qquad \Omega \vdash e' : \tau_i}{\Omega \vdash e[i]\ :=\ e' : \mathbf{1}}$$

$$\frac{\Omega \vdash e : \mathbf{ref}(\overline{\tau}) \qquad \tau_i = \sigma \qquad \Omega \vdash e_o : \sigma \qquad \Omega \vdash e_n : \sigma}{\Omega \vdash \mathsf{CAS}(e[i], e_o, e_n) : \mathbf{B}} \qquad \frac{\Omega \vdash e : \mathbf{1}}{\Omega \vdash \mathbf{fork}\ e : \mathbf{1}} \qquad \frac{\Omega \vdash e : \mu\alpha.\tau}{\Omega \vdash e : \tau[\mu\alpha.\tau/\alpha]}$$

$$\frac{\Omega \vdash e : \tau[\mu\alpha.\tau/\alpha]}{\Omega \vdash e : \mu\alpha.\tau}$$

1.3 Operational semantics

$$
\begin{array}{rcl}
K & ::= & [\,] \mid \textsf{if } K \textsf{ then } e \textsf{ else } e \mid K + e \mid v + K \mid K\ e \mid v\ K \\
& \mid & \textsf{inj}_i\ K \mid \textsf{case}(K, \textsf{inj}_1\ x \Rightarrow e, \textsf{inj}_2\ x \Rightarrow e) \mid \textsf{some}(K) \mid \textsf{case}(K, \textsf{null} \Rightarrow e, \textsf{some}(x) \Rightarrow e) \mid K\ \_ \\
& \mid & \textsf{new}\ (\overline{v}, K, \overline{e}) \mid K[i] \mid K[i] := e \mid v[i] := K \mid \mathsf{CAS}(K[i], e, e) \mid \mathsf{CAS}(v[i], K, e) \mid \mathsf{CAS}(v[i], v, K) \\
T & \in & \text{ThreadPool} \triangleq \mathbb{N} \xrightarrow{\text{fin}} \text{Exp} \\
u & ::= & (\overline{v}) \mid \textsf{inj}_i\ v \\
h & ::= & \cdot \mid h, \ell \mapsto u
\end{array}
$$

Primitive reductions $\boxed{h; e \hookrightarrow h'; e'}$

$$
\begin{array}{rcll}
h; n + m & \hookrightarrow & h; k & \text{when } k = n + m \\
h; \ell[i] & \hookrightarrow & h; v_i & \text{when } h(\ell) = (\overline{v}) \\
h; \ell[i] := v & \hookrightarrow & h[\ell[i] = v]; () & \text{when } \ell \in \text{dom}(h) \\
h; \mathsf{CAS}(\ell[i], v_o, v_n) & \hookrightarrow & h[\ell[i] = v_n]; \textsf{true} & \text{when } h(\ell)[i] = v_o \\
h; \mathsf{CAS}(\ell[i], v_o, v_n) & \hookrightarrow & h; \textsf{false} & \text{when } h(\ell)[i] \neq v_o \\
h; \textsf{case}(\ell, \textsf{inj}_1\ x \Rightarrow e_1, \textsf{inj}_2\ x \Rightarrow e_2) & \hookrightarrow & h; e_i[v/x] & \text{when } h(\ell) = \textsf{inj}_i\ v
\end{array}
$$

$$
\begin{array}{rcl}
h; \textsf{if true then } e_1 \textsf{ else } e_2 & \hookrightarrow & h; e_1 \\
h; \textsf{if false then } e_1 \textsf{ else } e_2 & \hookrightarrow & h; e_1 \\
h; \textsf{null} & \hookrightarrow & h; () \\
h; \textsf{some}(\ell) & \hookrightarrow & h; \ell \\
h; \textsf{case}((), \textsf{null} \Rightarrow e_1, \textsf{some}(x) \Rightarrow e_2) & \hookrightarrow & h; e_1 \\
h; \textsf{case}(\ell, \textsf{null} \Rightarrow e_1, \textsf{some}(x) \Rightarrow e_2) & \hookrightarrow & h; e_2[\ell/x] \\
h; \textsf{rec } f(x).e\ v & \hookrightarrow & h; e[\textsf{rec } f(x).e/f, v/x] \\
h; \textsf{inj}_i\ v & \hookrightarrow & h \uplus [\ell \mapsto \textsf{inj}_i\ v]; \ell \\
h; \Lambda.e\ \_ & \hookrightarrow & h; e \\
h; \textsf{new } (\overline{v}) & \hookrightarrow & h \uplus [\ell \mapsto (\overline{v})]; \ell
\end{array}
$$

Program reduction $\boxed{h; T \to h'; T'}$

$$
\frac{h; e \hookrightarrow h'; e'}{h; T \uplus [i \mapsto K[e]] \to h'; T \uplus [i \mapsto K[e']]} \qquad h; T \uplus [i \mapsto K[\textsf{fork } e]] \to h; T \uplus [i \mapsto K[()]] \uplus [j \mapsto e]
$$

1.4 Contextual approximation (refinement)

$$
\begin{array}{rcl}
\Omega \models e_1 \preceq e_2 : \tau & \triangleq & \forall C : (\Omega, \tau) \rightsquigarrow (\emptyset, \mathbf{N}).\ \forall i, j. \\
& & \text{if} \quad \emptyset; [i \mapsto C[e_1]] \to^* h_1; [i \mapsto n] \uplus T_1 \\
& & \text{then} \quad \emptyset; [j \mapsto C[e_2]] \to^* h_2; [j \mapsto n] \uplus T_2
\end{array}
$$

1.5 Derived forms

$$
\begin{array}{rcl}
\mathsf{getVal}(e) & \triangleq & \textbf{case}(e, \textbf{null} \Rightarrow \mathsf{diverge}, \textbf{some}(x) \Rightarrow x) \\
\\
\mathsf{list}(\tau) & \triangleq & \mu\alpha.\textbf{ref}_?(\alpha, \tau) \\
\mathsf{cons}(e, e') & \triangleq & \textbf{some}(\textbf{new}\ (e, e')) \\
\\
\textbf{let}\ x = e\ \textbf{in}\ e' & \triangleq & (\lambda\_.e')\ e \\
e; e' & \triangleq & \textbf{let}\ \_ = e\ \textbf{in}\ e' \\
\\
\mathsf{forkID}\ e & \triangleq & \textbf{let}\ x = \textbf{new none in fork}\ x\ :=\ \textbf{some}(e);\ x \\
\mathsf{join} & \triangleq & \textbf{rec}\ f(x).\ \textbf{case}(x[0], \_ \Rightarrow f(x), r \Rightarrow r) \\
\\
\mathsf{acq} & \triangleq & \textbf{rec}\ f(x).\ \textbf{if}\ \mathsf{CAS}(x[1], \textbf{false}, \textbf{true})\ \textbf{then}\ ()\ \textbf{else}\ f(x) \\
\mathsf{rel} & \triangleq & \lambda x.x[1]\ :=\ \textbf{false} \\
\mathsf{sync}(e)\ \{\ e'\ \} & \triangleq & \textbf{let}\ x = e\ \textbf{in}\ \mathsf{acq}(x);\ \textbf{let}\ r = e'\ \textbf{in}\ \mathsf{rel}(x);\ r
\end{array}
$$

2 The model

2.1 Standard resources

$$
\begin{array}{rcl}
\Sigma \in \text{StateSet} & \triangleq & \overline{\wp}(\text{Heap} \times \text{ThreadPool}) \quad \text{(nonempty, finite subsets)} \\
\eta \in \text{ImplSpec} & \triangleq & \text{Heap} \times \text{StateSet} \\
\Sigma_1 \otimes \Sigma_2 & \triangleq & \{\ h_1 \uplus h_2; T_1 \uplus T_2 \ \mid\ h_i; T_i \in \Sigma_i\ \} \quad \text{(defined only when } \textit{all} \text{ compositions are defined)} \\
(h_1, \Sigma_1) \otimes (h_2, \Sigma_2) & \triangleq & (h_1 \uplus h_2, \Sigma_1 \otimes \Sigma_2)
\end{array}
$$

2.2 Worlds and islands

$$
\begin{array}{rcl}
\text{STS} & \triangleq & \left\{\ \theta = (S, A, \rightsquigarrow, F) \ \middle|\ \rightsquigarrow \subseteq S \times S,\ F \in S \to \wp(A) \ \right\} \\
\text{World}_n & \triangleq & \left\{\ W = (k, \omega) \ \middle|\ k < n,\ \omega \in \mathbb{N} \xrightarrow{\text{fin}} \text{Island}_k \ \right\} \\
\text{Island}_n & \triangleq & \left\{\ (\theta, J, s, A) \ \middle|\ \begin{array}{l} \theta \in \text{STS},\ s \in \theta.S,\ A \subseteq \theta.A,\ A \# \theta.F(s), \\ J \in \theta.S \to \text{UWorld}_n \overset{\text{mon}}{\to} \wp(\text{ImplSpec}) \end{array} \ \right\} \\
\text{UWorld}_n & \triangleq & \{\ U \in \text{World}_n \ \mid\ U = |U|\ \} \\
\text{VRel}_n & \triangleq & \left\{\ V \in \text{UWorld}_n \overset{\text{mon}}{\to} \wp(\text{Val} \times \text{Val}) \ \right\}
\end{array}
$$

where $\overset{\text{mon}}{\to}$ denotes monotonic functions under the $\overset{\text{rely}}{\sqsubseteq}$ order.

$$
\begin{array}{rcl}
\lfloor \omega \rfloor_k & \triangleq & i \mapsto \lfloor \omega(i) \rfloor_k \\
\lfloor (\theta, J, s_0, A) \rfloor_k & \triangleq & (\theta, (s \mapsto I(s) \upharpoonright \text{UWorld}_k), s_0, A) \\
\theta \vdash (s, A) \rightsquigarrow (s', A') & \triangleq & s \rightsquigarrow_\theta s' \quad \wedge \quad \theta.F(s) \uplus A = \theta.F(s') \uplus A' \\
(\theta, J, s, A) \overset{\text{guar}}{\sqsubseteq} (\theta', J', s', A') & \triangleq & \theta = \theta',\ J = J',\ \theta \vdash (s, A) \rightsquigarrow^* (s', A') \\
\omega \overset{\text{guar}}{\sqsubseteq} \omega' & \triangleq & \forall i \in \text{dom}(\omega).\ \omega(i) \overset{\text{guar}}{\sqsubseteq} \omega'(i) \\
(k, \omega) \overset{\text{guar}}{\sqsubseteq} (k', \omega') & \triangleq & k \geq k' \quad \wedge \quad \lfloor \omega \rfloor_{k'} \overset{\text{guar}}{\sqsubseteq} \omega' \\
\text{frame}(k, \omega) & \triangleq & (k,\ i \mapsto \text{frame}(\omega(i))) \\
\text{frame}(\theta, J, s, A) & \triangleq & (\theta,\ J,\ s,\ \theta.A - (\theta.F(s) \cup A)) \\
W \overset{\text{rely}}{\sqsubseteq} W' & \triangleq & \text{frame}(W) \overset{\text{guar}}{\sqsubseteq} \text{frame}(W') \\
|(\theta, J, s, A)| & \triangleq & (\theta, J, s, \emptyset) \\
|(k, \omega)| & \triangleq & (k, i \mapsto |\omega(i)|) \\
(\theta, J, s, A) \otimes (\theta', J', s', A') & \triangleq & \begin{cases} (\theta, J, s, A \cup A') & \theta = \theta', s' = s, J' = J, A \pitchfork A' \\ \text{undefined} & \text{otherwise} \end{cases} \\
(k, \omega) \otimes (k', \omega') & \triangleq & \begin{cases} (k, \omega \otimes \omega') & k = k',\ \text{dom}(\omega) = \text{dom}(\omega'), \\ \text{undefined} & \text{otherwise} \end{cases} \\
W \# W' & \triangleq & W \otimes W' \text{ defined} \\
\mathcal{I}[\![(\theta, J, s, A)]\!]_W & \triangleq & J(s)(\triangleright |W|) \\
\mathcal{I}[\![\omega]\!]_W & \triangleq & \left\{\ \eta_{i_1} \otimes \cdots \otimes \eta_{i_n} \ \middle|\ \begin{array}{l} \text{dom}(\omega) = \{i_1, \ldots, i_n\} \wedge \\ 1 \leq j \leq n \implies \eta_{i_j} \in \mathcal{I}[\![W.\omega(i_j)]\!]_W \end{array} \ \right\} \\
h, \Sigma : W, \eta & \triangleq & W.k > 0 \implies \exists \eta_W \in \mathcal{I}[\![W.\omega]\!]_W.\ (h, \Sigma) = \eta \otimes \eta_W
\end{array}
$$

2.3 Assertions

$$
\begin{array}{rcl}
m & ::= & i \mid \mathsf{none} \\
\iota & ::= & (\theta, I, s, A) \text{ where } I \in \theta.S \to \text{Assert},\ s \in \theta.S,\ A \subseteq \theta.A,\ \theta.F(s) \# A \\
P & ::= & \mathsf{emp} \mid \triangleright P \mid \iota \mid v \mapsto_{\mathrm{I}} u \mid v \mapsto_{\mathrm{S}} u \mid i \rightarrowtail_{\mathrm{S}} e \mid T@m\ \langle x.\ P \rangle \\
 & \mid & P \Rightarrow P \mid P \wedge P \mid P \vee P \mid \exists x.P \mid \forall x.P \mid P * P \mid P \oplus P \mid \diamond P \mid \varphi \\
\varphi & ::= & v = v \mid (\!|P|\!)\ a\ (\!|x.\ Q|\!) \mid \langle P \rangle\ e\ \langle x.\ Q \rangle \mid \Omega \vdash e \preceq^{\mathcal{E}} e : \tau \mid v \preceq^{\mathcal{V}} v : \tau \\
\mathrm{inv}(P) & \triangleq & ((\{1\}, \emptyset, \emptyset, \lambda\_.\emptyset), \lambda\_.P, 1, \emptyset) \\
\Sigma \rightrightarrows \Sigma' & \triangleq & \forall \varsigma' \in \Sigma'.\ \exists \varsigma \in \Sigma.\ \varsigma \to^* \varsigma'
\end{array}
$$

$$
\begin{array}{rcl}
[\![I]\!]^{\rho}_{k} & : & \mathrm{dom}(I) \to \mathrm{UWorld}_k \overset{\mathrm{mon}}{\to} \wp(\mathrm{ImplSpec}) \\
[\![I]\!]^{\rho}_{k}(s)(U) & \triangleq & \{\ \eta \ \mid\ U, \eta \models^{\rho} I(s)\ \}
\end{array}
$$

Semantics of assertions (closed on term variables, may have free type variables closed by ρ):

$$
\begin{array}{lcl}
W, \eta \models^{\rho} \mathsf{emp} & \triangleq & W = |W|, \eta = (\emptyset, \{\emptyset; \emptyset\}) \\
W, \eta \models^{\rho} \triangleright P & \triangleq & W.k > 0 \implies \triangleright W, \eta \models^{\rho} P \\
W, \eta \models^{\rho} (\theta, I, s, A) & \triangleq & \exists i.\ W \overset{\mathrm{rely}}{\sqsupseteq} (W.k, [i \mapsto (\theta, [\![I]\!]^{\rho}_{W.k}, s, A)]) \\
W, \eta \models^{\rho} v \mapsto_{\mathrm{I}} u & \triangleq & \eta = ([v \mapsto u], \{\emptyset; \emptyset\}) \\
W, \eta \models^{\rho} v \mapsto_{\mathrm{S}} u & \triangleq & \eta = (\emptyset, \{[v \mapsto u]; \emptyset\}) \\
W, \eta \models^{\rho} i \rightarrowtail_{\mathrm{S}} e & \triangleq & \eta = (\emptyset, \{\emptyset; [i \mapsto e]\}) \\
W, \eta \models^{\rho} P \Rightarrow P' & \triangleq & \forall W' \overset{\mathrm{rely}}{\sqsupseteq} W.\ W', \eta \models^{\rho} P \text{ implies } W', \eta \models^{\rho} P' \\
W, \eta \models^{\rho} P \wedge P' & \triangleq & W, \eta \models^{\rho} P \text{ and } W, \eta \models^{\rho} P' \\
W, \eta \models^{\rho} P \vee P' & \triangleq & W, \eta \models^{\rho} P \text{ or } W, \eta \models^{\rho} P' \\
W, \eta \models^{\rho} \exists x.P & \triangleq & \exists v.\ W, \eta \models^{\rho} P[v/x] \\
W, \eta \models^{\rho} \forall x.P & \triangleq & \forall v.\ W, \eta \models^{\rho} P[v/x] \\
W, \eta \models^{\rho} P_1 * P_2 & \triangleq & W = W_1 \otimes W_2,\ \eta = \eta_1 \otimes \eta_2,\ W_i, \eta_i \models^{\rho} P_i \\
W, \eta \models^{\rho} P_1 \oplus P_2 & \triangleq & \eta.\Sigma = \Sigma_1 \cup \Sigma_2,\ W, (\eta.h, \Sigma_i) \models^{\rho} P_i \\
W, \eta \models^{\rho} \diamond P & \triangleq & \eta = (h, \Sigma),\ \forall \Sigma_F \# \Sigma.\ \exists \Sigma'.\ (\Sigma \otimes \Sigma_F) \rightrightarrows (\Sigma' \otimes \Sigma_F) \text{ and } W, (h, \Sigma') \models^{\rho} P \\
W, \eta \models^{\rho} \varphi & \triangleq & |W| \models^{\rho} \varphi \\
W_0, \eta \models^{\rho} T@m\ \langle x.\ Q \rangle & \triangleq & \forall W \overset{\mathrm{rely}}{\sqsupseteq} W_0,\ \eta_F \# \eta. \\
\quad \text{if } W.k > 0 \text{ and } h, \Sigma : W, \eta \otimes \eta_F \text{ then} & & \\
\quad\quad \text{if } h; T \to h'; T' \text{ then } \exists \Sigma', \eta', W' \overset{\mathrm{guar}}{\sqsupseteq} W. & & \\
\quad\quad\quad \Sigma \rightrightarrows \Sigma', \quad h', \Sigma' : W', \eta' \otimes \eta_F, \quad W'.k = W.k - 1, \quad W', \eta' \models^{\rho} T'@m\ \langle x.\ Q \rangle & & \\
\quad\quad \text{if } T = T_0 \uplus [m \mapsto v] \text{ then } \exists \Sigma', \eta', W' \overset{\mathrm{guar}}{\sqsupseteq} W. & & \\
\quad\quad\quad \Sigma \rightrightarrows \Sigma', \quad h, \Sigma' : W', \eta' \otimes \eta_F, \quad W'.k = W.k, \quad W', \eta' \models^{\rho} Q[v/x] * T_0@\mathsf{none}\ \langle x.\ \mathsf{tt} \rangle &
\end{array}
$$

Semantics of syntactically pure assertions:[1]

$$
\begin{array}{lll}
U \models^\rho \ v_1 = v_2 & \triangleq & v_1 = v_2 \\
U \models^\rho \ \cdot \overset{\text{rely}}{\sqsupseteq} U_0 & \triangleq & U \overset{\text{rely}}{\sqsupseteq} U_0 \\
U \models^\rho \ \langle P \rangle\ e\ \langle x.\ Q \rangle & \triangleq & \forall i.\ U \models^\rho P \Rightarrow [i \mapsto e] @ i\ \langle x.\ Q \rangle
\end{array}
$$

$$
\begin{array}{l}
U \models^\rho \ (\!|P|\!)\ a\ (\!|x.\ Q|\!) \ \triangleq\ \forall W \overset{\text{rely}}{\sqsupseteq} U, \eta, \eta_F \# \eta. \\
\quad \text{if } W.k > 0 \text{ and } \triangleright W, \eta \models^\rho P \text{ and } (\eta \otimes \eta_F).h; a \hookrightarrow h'; v \text{ then} \\
\qquad \exists \eta' \# \eta_F.\quad h' = (\eta' \otimes \eta_F).h, \quad (\eta \otimes \eta_F).\Sigma \rightrightarrows (\eta' \otimes \eta_F).\Sigma, \quad \triangleright W, \eta' \models^\rho Q[v/x]
\end{array}
$$

$U \models^\rho \ v_1 \preceq^{\mathcal{V}} v_2 : \tau_0 \ \triangleq$

τ_0	Requirements
τ_b	$v_1 = v_2,\ \vdash v_i : \tau_b$ for $\tau_b \in \{\mathbf{1}, \mathbf{B}, \mathbf{N}\}$
α	$\overline{(v_1, v_2) \in \rho(\alpha)(U)}$
$\tau \to \tau'$	$\overline{v_i = \mathbf{rec}\ f\ x.e_i},\ U \models^\rho \triangleright(x : \tau \vdash e_1[v_1/f] \preceq^{\mathcal{E}} e_2[v_2/f] : \tau')$
$\forall \alpha.\tau$	$\overline{v_i = \Lambda.e_i},\qquad U \models^\rho \triangleright(\alpha \vdash e_1 \preceq^{\mathcal{E}} e_2 : \tau)$
$\mu\alpha.\tau$	$U \models^\rho v_1 \preceq^{\mathcal{V}} v_2 : \tau[\mu\alpha.\tau/\alpha]$
$\mathbf{ref}_?(\overline{\tau})$	$U \models^\rho v_1 \preceq^{\mathcal{V}} v_2 : \mathbf{1} \ \vee\ \overline{v_1 \preceq^{\mathcal{V}} v_2 : \mathbf{ref}(\overline{\tau})}$
$\mathbf{ref}(\overline{\tau})$	$U \models^\rho \mathrm{inv}(\exists \overline{x}, \overline{y}.\ \bigwedge \overline{x \preceq^{\mathcal{V}} y : \tau} \wedge (v_1 \mapsto_{\mathrm{I}} (\overline{x}) * v_2 \mapsto_{\mathrm{S}} (\overline{y})))$
$\tau_1 + \tau_2$	$U \models^\rho \exists x, y.\ x \preceq^{\mathcal{V}} y : \tau_i \wedge \mathrm{inv}(v_1 \mapsto_{\mathrm{I}} \mathbf{inj}_i\ x * v_2 \mapsto_{\mathrm{S}} \mathbf{inj}_i\ y)$

$$
\begin{array}{lll}
U \models^\rho \ (\ \cdot\ ;\ \cdot\ \vdash e_1 \preceq^{\mathcal{E}} e_2 : \tau) & \triangleq & \forall K, j.\ U \models^\rho \langle j \rightarrowtail_{\mathrm{S}} K[e_2] \rangle\ e_1\ \langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau \wedge j \rightarrowtail_{\mathrm{S}} K[x_2] \rangle \\
U \models^\rho \ (\ \cdot\ ; x : \tau', \Gamma \vdash e_1 \preceq^{\mathcal{E}} e_2 : \tau) & \triangleq & U \models^\rho \forall x_1, x_2.\ (x_1 \preceq^{\mathcal{V}} x_2 : \tau') \Rightarrow (\Gamma \vdash e_1[x_1/x] \preceq^{\mathcal{E}} e_2[x_2/x] : \tau) \\
U \models^\rho \ (\alpha, \Delta;\ \Gamma \vdash e_1 \preceq^{\mathcal{E}} e_2 : \tau) & \triangleq & \forall V.\ U \models^{\rho[\alpha \mapsto V]} \Delta; \Gamma \vdash e_1 \preceq^{\mathcal{E}} e_2 : \tau
\end{array}
$$

An assertion P is *token-pure* if $W, \eta \models^\rho P$ iff $|W|, \eta \models^\rho P$.

The set World $\times$ ImplSpec is a preorder when ordered by $(w, \eta) \sqsupseteq (w', \eta')$ iff $w \overset{\text{rely}}{\sqsupseteq} w' \wedge \eta = \eta'$. Since ImplSpec is a partial commutative monoid and World is a commutative semi-group with a unit element for each element, we have:

Lemma 1. The set $P^{\uparrow}$(World $\times$ ImplSpec) of upwards-closed subsets of World $\times$ ImplSpec is a complete BI-algebra.

Hence we get a model of intuitionistic BI logic and, indeed, the interpretation of the logical connectives is as detailed in the above table showing the semantics of assertions.

[1] If P is impure, $U \models^\rho P$ is short for $\forall \eta.\ U, \eta \models^\rho P$.

3 Soundness

3.1 Basic Properties

Lemma 2 (Rely-guarantee Preorders)**.** The relations $\overset{\text{rely}}{\sqsubseteq}$ and $\overset{\text{guar}}{\sqsubseteq}$ are preorders.

Lemma 3 (Rely-closure of Assertions)**.** $W, \eta \models^\rho P$ and $W' \overset{\text{rely}}{\sqsupseteq} W$ implies $W', \eta \models^\rho P$.

Lemma 4. $|W| \otimes W = W$.

Lemma 5. If $W \overset{\text{rely}}{\sqsubseteq} W'$ then $|W| \overset{\text{rely}}{\sqsubseteq} |W'|$.

Lemma 6 (Rely Decomposition)**.** If $W_1 \otimes W_2 \overset{\text{rely}}{\sqsubseteq} W'$ then there are W_1' and W_2' with $W' = W_1' \otimes W_2'$, $W_1 \overset{\text{rely}}{\sqsubseteq} W_1'$ and $W_2 \overset{\text{rely}}{\sqsubseteq} W_2'$.

Lemma 7 (Token Framing)**.** If $W \overset{\text{guar}}{\sqsubseteq} W'$ and $W \otimes W_f$ is defined then there exists some $W_f' \overset{\text{rely}}{\sqsupseteq} W_f$ such that $W' \otimes W_f'$ is defined and $W \otimes W_f \overset{\text{guar}}{\sqsubseteq} W' \otimes W_f'$.

Lemma 8. If $h, \Sigma : W, \eta$ then $h, \Sigma : W \otimes W', \eta$.

Lemma 9. If $h, \Sigma : W \otimes W', \eta$ then $h, \Sigma : W, \eta$.

Lemma 10. If $W.k > 0$ then $\triangleright W \overset{\text{guar}}{\sqsupseteq} W$ and $\triangleright W \overset{\text{rely}}{\sqsupseteq} W$.

Lemma 11. If $W.k > 0$ then $|\triangleright W| = \triangleright |W|$.

Lemma 12 (Later Satisfaction)**.** If $W.k > 0$ and $h, \Sigma : W, \eta$ then $h, \Sigma : \triangleright W, \eta$.

Lemma 13. $\rightrightarrows$ is transitive.

3.2 Constructions with Threadpool Triples

Lemma 14 (Framing)**.** $W, \eta \models^\rho T@m \langle x.\, Q \rangle$ and $W_f, \eta_f \models^\rho R$ with $W \# W_f$, $\eta \# \eta_f$ gives

$$W \otimes W_f, \eta \otimes \eta_f \models^\rho T@m \langle x.\, Q * R \rangle.$$

Proof. The proof proceeds by induction on $W.k$.

Case $\boxed{W.k = 0}$

1. $(W \otimes W_f).k = W.k = 0$.

Case $\boxed{W.k > 0}$

2. Let $W' \overset{\text{rely}}{\sqsupseteq} W \otimes W_f, \eta_f' \# \eta \otimes \eta_f$.
3. Write $W' = W_1' \otimes W_2'$ with $W_1' \overset{\text{rely}}{\sqsupseteq} W, W_2' \overset{\text{rely}}{\sqsupseteq} W_f$ by Lem. 6.
4. Suppose $(W_1' \otimes W_2').k > 0$.
5. $W_1'.k = (W_1' \otimes W_2').k > 0$.
6. Suppose $h, \Sigma : W_1' \otimes W_2', \eta \otimes \eta_f \otimes \eta_f'$.
7. $h, \Sigma : W_1', \eta \otimes \eta_f \otimes \eta_f'$ by Lem. 9.

Case $\boxed{h; T \to h'; T'}$

8. Pick $\Sigma', \eta',$ and W_1'' with $W_1'' \overset{\text{guar}}{\sqsupseteq} W_1'$, $\Sigma \rightrightarrows \Sigma'$, $h', \Sigma' : W_1'', \eta' \otimes \eta_f \otimes \eta_f'$, $W_1''.k = W_1'.k - 1$, $W_1'', \eta' \models^\rho T'@m \left\langle x.\ Q \right\rangle$ — by assumption.
9. Pick $W_2'' \overset{\text{rely}}{\sqsupseteq} W_2'$ with $W_1'' \otimes W_2'' \overset{\text{guar}}{\sqsupseteq} W_1' \otimes W_2'$ — by Lem. 7.
10. $h', \Sigma' : W_1'' \otimes W_2'', \eta' \otimes \eta_f \otimes \eta_f'$ — by Lem. 8.
11. $(W_1'' \otimes W_2'').k = W_1''.k = W_1'.k - 1 = (W_1' \otimes W_2').k - 1$.
12. $W_2'', \eta_f \models^\rho R$.
13. $W_1'' \otimes W_2'', \eta' \otimes \eta_f \models^\rho T'@m \left\langle x.\ Q * R \right\rangle$ — by induction hypothesis.

Case $\boxed{m = i \text{ and } T = T_0 \uplus [i \mapsto v]}$

14. Pick $\Sigma', \eta',$ and W_1'' with $W_1'' \overset{\text{guar}}{\sqsupseteq} W_1'$, $\Sigma \rightrightarrows \Sigma'$, $h, \Sigma' : W_1'', \eta' \otimes \eta_f \otimes \eta_f'$, $W_1''.k = W_1'.k$, $W_1'', \eta' \models^\rho Q[v/x] * T_0@\mathsf{none} \left\langle x.\ \mathsf{tt} \right\rangle$ — by assumption.
15. Pick $W_2'' \overset{\text{rely}}{\sqsupseteq} W_2'$ with $W_1'' \otimes W_2'' \overset{\text{guar}}{\sqsupseteq} W_1' \otimes W_2'$ — by Lem. 7.
16. $h', \Sigma' : W_1'' \otimes W_2'', \eta' \otimes \eta_f \otimes \eta_f'$ — by Lem. 8.
17. $(W_1'' \otimes W_2'').k = W_1''.k = W_1'.k = (W_1' \otimes W_2').k$.
18. $W_2'', \eta_f \models^\rho R$.
19. $W_1'' \otimes W_2'', \eta' \otimes \eta_f \models^\rho Q[v/x] * R * T_0@\mathsf{none} \left\langle x.\ \mathsf{tt} \right\rangle$.

□

Corollary 1 (Precondition Extension)**.** $W, \eta \models^\rho T@m \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau \wedge j \rightarrowtail_{\mathrm{S}} K[x_2] \right\rangle$ together with $W_f \# W$ gives $W \otimes W_f, \eta \models^\rho T@m \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau \wedge j \rightarrowtail_{\mathrm{S}} K[x_2] \right\rangle$.

Corollary 2 (Postcondition Strengthening)**.** If we have $W, \eta \models^\rho T@m \left\langle x.\ Q \right\rangle$ then $W, \eta \models^\rho T@m \left\langle x.\ Q \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W| \right\rangle$ holds too.

Lemma 15 (Parallel Composition)**.** If we have $W_1, \eta_1 \models^\rho T_1@m_1 \left\langle x.\ Q_1 \right\rangle$ and $W_2, \eta_2 \models^\rho T_2@m_2 \left\langle x.\ Q_2 \right\rangle$ with $W_1 \# W_2$, $\eta_1 \# \eta_2$, $T_1 \# T_2$ and $m_1 \neq \mathsf{none} \Rightarrow m_1 \in \mathrm{dom}(T_1)$ then we also have

$$W_1 \otimes W_2, \eta_1 \otimes \eta_2 \models^\rho T_1 \uplus T_2@m_1 \left\langle x.\ Q_1 \right\rangle.$$

Proof. The proof proceeds by induction on the measure $M(W_1, m_1)$ defined by

$$M(W, m) = \begin{cases} W.k & m = \mathsf{none} \\ W.k + 1 & m \neq \mathsf{none}. \end{cases}$$

Case $\boxed{M(W_1, m_1) = 0}$

1. $(W_1 \otimes W_2).k = W_1.k = 0$.

Case $\boxed{M(W_1, m_1) > 0}$

2. Let $W' \overset{\text{rely}}{\sqsupseteq} W_1 \otimes W_2, \eta_f \# \eta_1 \otimes \eta_2$.
3. Write $W' = W_1' \otimes W_2'$ with $W_1' \overset{\text{rely}}{\sqsupseteq} W_1, W_2' \overset{\text{rely}}{\sqsupseteq} W_2$ by Lem. 6.
4. Suppose $(W_1' \otimes W_2').k > 0$.
5. $W_1'.k = (W_1' \otimes W_2').k > 0$.
6. Suppose $h, \Sigma : W_1' \otimes W_2', \eta_1 \otimes \eta_2 \otimes \eta_f$.
7. $h, \Sigma : W_1', \eta_1 \otimes \eta_2 \otimes \eta_f$ by Lem. 9.

Case $\boxed{h; T_1 \uplus T_2 \to h'; T'}$

8. Write $T' = T_1' \uplus T_2$ WLOG.
9. $h; T_1 \to h'; T_1'$ by nondeterminism of fork.
10. Pick $\Sigma', \eta_1',$ and W_1'' with by assumption.
    $W_1'' \overset{\text{guar}}{\sqsupseteq} W_1'$,
    $\Sigma \rightrightarrows \Sigma'$,
    $W_1''.k = W_1'.k - 1$,
    $h', \Sigma' : W_1'', \eta_1' \otimes \eta_2 \otimes \eta_f$,
    $W_1'', \eta_1' \models^\rho T_1' @ m_1 \left\langle x_1.\ Q_1 \right\rangle$
11. Pick $W_2'' \overset{\text{rely}}{\sqsupseteq} W_2'$ with $W_1'' \otimes W_2'' \overset{\text{guar}}{\sqsupseteq} W_1' \otimes W_2'$ by Lem. 7.
12. $(W_1'' \otimes W_2'').k = W_1''.k = W_1'.k - 1 = (W_1' \otimes W_2').k - 1$.
13. $h', \Sigma' : W_1'' \otimes W_2'', \eta_1' \otimes \eta_2 \otimes \eta_f$ by Lem. 8.
14. $W_2'', \eta_2 \models^\rho T_2 @ m_2 \left\langle x_2.\ Q_2 \right\rangle$ by assumption.
15. $W_1'' \otimes W_2'', \eta_1' \otimes \eta_2 \models^\rho T_1' \uplus T_2 @ m_1 \left\langle x_1.\ Q_1 \right\rangle$ by induction hypothesis.

Case $\boxed{T_1 * T_2 = T_0 \uplus [m_1 \mapsto v_1]}$

16. $m_1 \in \mathrm{dom}(T_1)$ by assumption.
17. Write $T_1 = T_1' \uplus [m_1 \mapsto v_1]$.
18. Pick $\Sigma', \eta_1',$ and W_1'' with by assumption.
    $W_1'' \overset{\text{guar}}{\sqsupseteq} W_1'$,
    $\Sigma \rightrightarrows \Sigma'$,
    $W_1''.k = W_1'.k$,
    $h, \Sigma' : W_1'', \eta_1' \otimes \eta_2 \otimes \eta_f$,
    $W_1'', \eta_1' \models^\rho Q_1[v_1/x_1] * T_1' @ \mathsf{none} \left\langle x_1.\ \mathsf{tt} \right\rangle$
19. Pick $W_2'' \overset{\text{rely}}{\sqsupseteq} W_2'$ with $W_1'' * W_2'' \overset{\text{guar}}{\sqsupseteq} W_1' * W_2'$ by Lem. 7.
20. $(W_1'' * W_2'').k = W_1''.k = W_1'.k = (W_1' * W_2').k$.
21. $h, \Sigma' : W_1'' * W_2'', \eta_1' * \eta_2 * \eta_f$ by Lem. 8.
22. $W_2'', \eta_2 \models^\rho T_2 @ m_2 \left\langle x_2.\ Q_2 \right\rangle$. by assumption.
23. $W_1'' * W_2'', \eta_1' * \eta_2 \models^\rho Q[v_1/x_1] * T_1' \uplus T_2 @ \mathsf{none} \left\langle x_1.\ \mathsf{tt} \right\rangle$. by induction hypothesis.

□

Lemma 16 (Sequential Composition)**.** If we have $W, \eta \models^\rho [i \mapsto e] \uplus T@i \left\langle x.\ Q \right\rangle$ and for all v and any W', η' with $W', \eta' \models Q[v/x]$ we have $W', \eta' \models^\rho [i \mapsto K[v]]@i \left\langle x.\ R \right\rangle$ then

$$W, \eta \models^\rho [i \mapsto K[e]] \uplus T@i \left\langle x.\ R \right\rangle.$$

Proof. The proof proceeds by induction on $W.k$; the case $W.k = 0$ is trivial so we assume $W.k > 0$. We branch on the structure of e:

1. Let $W' \overset{\text{rely}}{\sqsupseteq} W, \eta_f \# \eta$.
2. Suppose $W'.k > 0$.
3. Suppose $h, \Sigma : W', \eta \otimes \eta_f$.

Case $\boxed{e = v}$

4. Pick $\Sigma', \eta',$ and W'' with — by assumption.
   $W'' \overset{\text{guar}}{\sqsupseteq} W'$,
   $\Sigma \rightrightarrows \Sigma'$,
   $h, \Sigma' : W'', \eta' \otimes \eta_f$,
   $W''.k = W'.k$,
   $W'', \eta' \models^\rho Q[v/x] * T@\mathsf{none} \left\langle x.\ \mathsf{tt} \right\rangle$
5. Write $W'' = W''_1 \otimes W''_2$ and $\eta' = \eta'_1 \otimes \eta'_2$. with
   $W''_1, \eta'_1 \models^\rho Q[v/x]$,
   $W''_2, \eta'_2 \models^\rho T@\mathsf{none} \left\langle x.\ \mathsf{tt} \right\rangle$.
6. $W''_1, \eta'_1 \models^\rho [i \mapsto K[v]]@i \left\langle x.\ R \right\rangle$ — by assumption.
7. $W'', \eta' \models^\rho [i \mapsto K[v]] \uplus T@i \left\langle x.\ R \right\rangle$ — by Lem. 15.

Case $\boxed{K[v] = v'}$

8. Pick $\Sigma'', \eta'',$ and W''' with — by (7).
   $W''' \overset{\text{guar}}{\sqsupseteq} W''$,
   $\Sigma' \rightrightarrows \Sigma''$,
   $h, \Sigma'' : W''', \eta'' \otimes \eta_f$,
   $W'''.k = W''.k$,
   $W''', \eta'' \models^\rho R[v'/x] * T@\mathsf{none} \left\langle x.\ \mathsf{tt} \right\rangle$
9. $\Sigma \rightrightarrows \Sigma''$.
10. $W'''.k = W''.k = W'.k$.

Case $\boxed{h; [i \mapsto K[v]] \uplus T \to h'; T'}$

11. Pick $\Sigma'', \eta'',$ and W''' with — by (7).
    $W''' \overset{\text{guar}}{\sqsupseteq} W''$,
    $\Sigma' \rightrightarrows \Sigma''$,
    $h', \Sigma'' : W''', \eta'' \otimes \eta_f$,
    $W'''.k = W''.k - 1$,
    $W''', \eta'' \models^\rho T'@i \left\langle x.\ R \right\rangle$
12. $\Sigma \rightrightarrows \Sigma''$.
13. $W'''.k = W''.k - 1 = W'.k - 1$.

Case $\boxed{e \neq v}$

14. Suppose $h; [i \mapsto K[e]] \uplus T \to h'; [i \mapsto K[e']] \uplus T'$.
15. $h; [i \mapsto e] \uplus T \to h'; [i \mapsto e'] \uplus T'$.
16. Pick $\Sigma', \eta',$ and W'' with — by assumption.
    $W'' \overset{\text{guar}}{\sqsupseteq} W'$,
    $\Sigma \rightrightarrows \Sigma'$,
    $h', \Sigma' : W'', \eta' \otimes \eta_f$,
    $W''.k = W'.k - 1$,
    $W'', \eta' \models^\rho [i \mapsto e'] \uplus T' @ i \left\langle x.\ Q \right\rangle$
17. $W'', \eta' \models^\rho [i \mapsto K[e']] \uplus T' @ i \left\langle x.\ R \right\rangle$ — by induction hypothesis.

□

3.3 Congruence

Lemma 17 (Soundness Shortcut)**.** $\Delta; \Gamma \models e_1 \preceq e_2 : \tau$ is equivalent to

$$\begin{aligned}
&\forall U\ \forall \rho : \Delta \to \mathrm{VRel}\ \forall \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}. \\
&\quad \big[\forall x \in \mathrm{dom}(\Gamma).\ U \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)\big] \\
&\qquad \Longrightarrow \\
&\quad \big[\forall K, j, i.\ U, (\emptyset, \{\emptyset; [j \mapsto K[e_2[\gamma_2/\Gamma]]]\}) \models^\rho \\
&\qquad [i \mapsto e_1[\gamma_1/\Gamma]] @ i \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau \wedge j \rightarrowtail_{\mathrm{S}} K[x_2] \right\rangle\big].
\end{aligned}$$

3.3.1 New

Lemma 18.

$$\frac{\Delta; \Gamma \models e_i \preceq f_i : \tau_i}{\Delta; \Gamma \models \mathsf{new}\ \overline{e} \preceq \mathsf{new}\ \overline{f} : \mathsf{ref}(\overline{\tau})}$$

Proof.

1. Let $U, \rho : \Delta \to \mathrm{VRel}, \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}$.
2. Suppose $\forall x \in \mathrm{dom}(\Gamma).\ U \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
3. Write $e_i' = e_i[\gamma_1/\Gamma], f_i' = f_i[\gamma_2/\Gamma]$.
4. Let K, i, j.
5. Write $\eta = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{new}\ \overline{f'}]]\})$.
6. Write $Q = \exists y.\ x \preceq^{\mathcal{V}} y : \mathsf{ref}(\overline{\tau}) \wedge j \rightarrowtail_{\mathrm{S}} K[y]$.
7. Suffices to show $U, \eta \models^\rho [i \mapsto \mathsf{new}\ \overline{e'}] @ i \left\langle x.\ Q \right\rangle$. — by Lem. 17.

Let $M = |\mathsf{ref}(\overline{\tau})|$. We now proceed to make a claim: for any $0 \leq m \leq M$ it sufices to prove

$$U', \eta_m \models^\rho [i \mapsto \mathsf{new}\ v_1, \ldots, v_m, e'_{m+1}, \ldots, e'_M] @ i \left\langle x.\ Q \right\rangle,$$

for all $U' \overset{\text{rely}}{\sqsupseteq} U$ and all $U' \models^\rho v_1 \preceq^{\mathcal{V}} w_1 : \tau_1, \ldots, U' \models^\rho v_m \preceq^{\mathcal{V}} w_m : \tau_m$, where $\eta_m = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{new}\ w_1, \ldots, w_m, f'_{m+1}, \ldots, f'_M]]\})$. We prove the claim by induction on m; the case

$m = 0$ was proved above. So, suppose the claim holds for $0 \leq m < M$ and assume that we know that

$$U'', \eta_{m+1} \models^{\rho} [i \mapsto \textsf{new}\ v_1, \ldots, v_{m+1}, e'_{m+2}, \ldots, e'_M]@i\ \langle x.\ Q \rangle,$$

holds for all for all $U'' \overset{\text{rely}}{\sqsupseteq} U$ and all $U'' \models^{\rho} v_1 \preceq^{\mathcal{V}} w_1 : \tau_1, \ldots, U'' \models^{\rho} v_{m+1} \preceq^{\mathcal{V}} w_{m+1} : \tau_{m+1}$, where $\eta_{m+1} = (\emptyset, \{\emptyset; [j \mapsto K[\textsf{new}\ w_1, \ldots, w_{m+1}, f'_{m+2}, \ldots, f'_M]]\})$. In the interest of applying the induction hypothesis, we pick arbitrary $U' \overset{\text{rely}}{\sqsupseteq} U$ and $U' \models^{\rho} v_1 \preceq^{\mathcal{V}} w_1 : \tau_1, \ldots, U' \models^{\rho} v_m \preceq^{\mathcal{V}} w_m : \tau_m$. By induction, it will suffice for the claim to prove that

$$U', \eta_m \models^{\rho} [i \mapsto \textsf{new}\ v_1, \ldots, v_m, e'_{m+1}, \ldots, e'_M]@i\ \langle x.\ Q \rangle,$$

holds, where $\eta_m = (\emptyset, \{\emptyset; [j \mapsto K[\textsf{new}\ w_1, \ldots, w_m, f'_{m+1}, \ldots, f'_M]]\})$. Now, by assumption and Lemma 17 and Corollary 2 we have

$$U', \eta_m \models^{\rho} [i \mapsto e'_{m+1}]@i\ \langle x_{m+1}.\ Q_{m+1} \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U' \rangle,$$

where $Q_{m+1} = \exists y_{m+1}.\ x_{m+1} \preceq^{\mathcal{V}} y_{m+1} : \tau_{m+1} \wedge j \rightarrowtail_{\mathrm{S}} K[\textsf{new}\ w_1, \ldots, w_m, y'_{m+1}, \ldots, f'_M]$. Now, let v_{m+1} be arbitrary and take $W'', \eta'' \models^{\rho} Q_{m+1}[v_{m+1}/x_{m+1}] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U'$ and by an application of Lemma 16 we have the claim if we can show

$$W'', \eta'' \models^{\rho} [i \mapsto \textsf{new}\ v_1, \ldots, v_{m+1}, e'_{m+2}, \ldots, e'_M]@i\ \langle x.\ Q \rangle.$$

Luckily, we can pick w_{n+1} such that we have $|W''| \models^{\rho} v_{m+1} \preceq^{\mathcal{V}} w_{m+1} : \tau_{m+1}, \eta'' = (\emptyset, \{\emptyset; [j \mapsto K[\textsf{new}\ w_1, \ldots, w_{m+1}, f'_{m+2}$ and $|W''| \overset{\text{rely}}{\sqsupseteq} U'$ and we can apply our original assumption. After this detour, we proceed with the proof proper:

8. Let $U' \overset{\text{rely}}{\sqsupseteq} U$.
9. Let $\bigwedge \overline{U' \models^{\rho} v \preceq^{\mathcal{V}} w : \tau}$.
10. Write $\eta' = (\emptyset, \{\emptyset; [j \mapsto K[\textsf{new}\ \overline{w}]]\})$.
11. Suffices to show $U', \eta' \models^{\rho} [i \mapsto \textsf{new}\ \overline{v}]@i\ \langle x.\ Q \rangle$.
12. Let $W' \overset{\text{rely}}{\sqsupseteq} U', \eta_f \# \eta'$.
13. Suppose $W'.k > 0$ and $h, \Sigma : W', \eta' \otimes \eta_f$.
14. $h; [i \mapsto \textsf{new}\ \overline{v}] \rightarrow h \uplus [\ell_1 \mapsto \overline{v}]; [i \mapsto \ell_1]$.
15. Pick ℓ_2 with $\forall h_2; T_2 \in \Sigma.\ \ell_2 \notin \mathrm{dom}(h_2)$.
16. Write $\Sigma = \{\emptyset; [j \mapsto K[\textsf{new}\ \overline{w}]]\} \otimes \Sigma_0$.
17. Write $\Sigma' = \{[\ell_2 \mapsto \overline{w}]; [j \mapsto K[\ell_2]]\} \otimes \Sigma_0$.
18. $\Sigma \rightrightarrows \Sigma'$.
19. Write $\eta'' = (\emptyset, \{\emptyset; [j \mapsto K[\ell_2]]\})$.
20. Pick $n \notin \mathrm{dom}(W'.\omega)$.
21. Write $P = \exists \overline{x}, \overline{y}.\ \bigwedge \overline{x \preceq^{\mathcal{V}} y : \tau} \wedge (\ell_1 \mapsto_{\mathrm{I}} (\overline{x}) * \ell_2 \mapsto_{\mathrm{S}} (\overline{y}))$.
22. Write $\iota = ((\{1\}, \emptyset, \emptyset, \lambda\_.\emptyset), [\![\lambda\_.P]\!]^{\rho}_{W'k}, 1, \emptyset)$.
23. Write $W'' = (W'.k, W.\omega \uplus [n \mapsto \iota])$.
24. $|W''| \models^{\rho} \ell_1 \preceq^{\mathcal{V}} \ell_2 : \textsf{ref}(\overline{\tau})$.
25. $\triangleright W'' \overset{\text{guar}}{\sqsupseteq} W'$.
26. $\triangleright |W''| \overset{\text{rely}}{\sqsupseteq} \triangleright |W'|$.

27. $\triangleright|W''| \overset{\text{rely}}{\sqsupseteq} U'$.
28. $h \uplus [\ell_1 \mapsto \overline{v}], \Sigma' : W'', \eta'' \otimes \eta_f$.
29. $h \uplus [\ell_1 \mapsto \overline{v}], \Sigma' : \triangleright W'', \eta'' \otimes \eta_f$ by Lem. 12 .
30. $\triangleright W'', \eta'' \models^{\rho} [i \mapsto \ell_1]@i \langle x.\, Q \rangle$.

□

3.3.2 Fork

Lemma 19.

$$\frac{\Delta; \Gamma \models e_1 \preceq e_2 : \mathbf{1}}{\Delta; \Gamma \models \mathsf{fork}\ e_1 \preceq \mathsf{fork}\ e_2 : \mathbf{1}}$$

Proof.

1. Let $U, \rho : \Delta \to \mathrm{VRel}, \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}$.
2. Suppose $\forall x \in \mathrm{dom}(\Gamma).\ U \models^{\rho} \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
3. Write $e_1' = e_1[\gamma_1/\Gamma], e_2' = e_2[\gamma_2/\Gamma]$.
4. Let K, i, j.
5. Write $\eta = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{fork}\ e_2']]\})$.
6. Write $Q = \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \mathbf{1} \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]$.
7. Suffices to show $U, \eta \models^{\rho} [i \mapsto \mathsf{fork}\ e_1']@i \langle x_1.\, Q \rangle$. by Lem. 17.
8. Let $W \overset{\text{rely}}{\sqsupseteq} U, \eta_f \# \eta$.
9. Suppose $W.k > 0$ and $h, \Sigma : W, \eta \otimes \eta_f$.
10. $h; [i \mapsto \mathsf{fork}\ e_1'] \to h; [i \mapsto ()] \uplus [i' \mapsto e_1']$.
11. Pick j' with $\forall h'; T' \in \Sigma.\ j' \notin \mathrm{dom}(T')$.
12. Write $\Sigma = \{\emptyset; [j \mapsto K[\mathsf{fork}\ e_2']]\} \otimes \Sigma_0$.
13. Write $\Sigma' = \{\emptyset; [j \mapsto K[()]] \uplus [j' \mapsto e_2']\} \otimes \Sigma_0$.
14. $\Sigma \rightrightarrows \Sigma'$.
15. Write $\eta' = (\emptyset, \{\emptyset; [j \mapsto K[()]]\})$.
16. Write $\eta'' = (\emptyset, \{\emptyset; [j' \mapsto e_2']\})$.
17. $h, \Sigma' : W, \eta' \otimes \eta'' \otimes \eta_f$.
18. $h, \Sigma' : \triangleright W, \eta' \otimes \eta'' \otimes \eta_f$ by Lem. 12.
19. $\triangleright W, \eta' \models^{\rho} [i \mapsto ()]@i \langle x_1.\, Q \rangle$.
20. $|\triangleright W|, \eta'' \models^{\rho} [i' \mapsto e_1']@i' \langle x_1.\, \mathsf{tt} \rangle$ by assumption and Lem. 17. .
21. $\triangleright W, \eta' \otimes \eta'' \models^{\rho} [i \mapsto ()] \uplus [i' \mapsto e_1']@i \langle x_1.\, Q \rangle$ by Lem. 15.

□

3.3.3 Function Application and Abstraction

Lemma 20. For $U.k \neq 0$ we have $U \models^\rho \textbf{rec}\ f(x).e_1 \preceq^{\mathcal{V}} \textbf{rec}\ f(x).e_2 : \tau_1 \to \tau_2$ equivalent to

$$\begin{aligned}
&\forall w_1, w_2.\ \forall U' \overset{\text{rely}}{\sqsupseteq} \triangleright U.\\
&\quad \left[U' \models^\rho w_1 \preceq^{\mathcal{V}} w_2 : \tau_1\right]\\
&\quad\quad \Longrightarrow\\
&\quad \left[\forall K, j, i.\ U', (\emptyset, \{\emptyset; [j \mapsto K[e_2[\textbf{rec}\ f(x).e_2/f, w_2/x]]]\}) \models^\rho\right.\\
&\quad\quad \left.[i \mapsto e_1[\textbf{rec}\ f(x).e_1/f, w_1/x]]@i \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau_2 \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]\right\rangle\right].
\end{aligned}$$

Lemma 21.

$$\frac{\Delta; \Gamma \models e_1 \preceq e_2 : \tau_1 \to \tau_2 \qquad \Delta; \Gamma \models f_1 \preceq f_2 : \tau_1}{\Delta; \Gamma \models e_1\ f_1 \preceq e_2\ f_2 : \tau_2}$$

Proof.

1. Let $U, \rho : \Delta \to \mathrm{VRel}, \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}$.
2. Suppose $\forall x \in \mathrm{dom}(\Gamma).\ U \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
3. Write $e_1' = e_1[\gamma_1/\Gamma], e_2' = e_2[\gamma_2/\Gamma]$.
4. Write $f_1' = f_1[\gamma_1/\Gamma], f_2' = f_2[\gamma_2/\Gamma]$.
5. Let K, i, j.
6. Write $\eta = (\emptyset, \{\emptyset; [j \mapsto K[e_2'\ f_2']]\})$.
7. Write $Q = \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau_2 \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]$.
8. Suffices to show $U, \eta \models^\rho [i \mapsto e_1'\ f_1']@i \left\langle x_1.\ Q\right\rangle$.
   by Lem. 17.
9. Write $Q' = \exists x_2'.\ x_1' \preceq^{\mathcal{V}} x_2' : \tau_1 \to \tau_2 \wedge j \rightarrowtail_{\mathrm{S}} K[x_2'\ f_2']$.
10. $U, \eta \models^\rho [i \mapsto e_1']@i \left\langle x_1'.\ Q'\right\rangle$. by assumption and Lem. 17.
11. $U, \eta \models^\rho [i \mapsto e_1']@i \left\langle x_1'.\ Q' \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U\right\rangle$ by Cor. 2.
12. Let $v_1' \in \mathrm{Val}$.
13. Let W', η' with $W', \eta' \models^\rho Q'[v_1'/x_1'] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U$.
14. Suffices to show $W', \eta' \models^\rho [i \mapsto v_1'\ f_1']@i \left\langle x_1.\ Q\right\rangle$
    by Lem. 16.
15. Suffices to show $|W'|, \eta' \models^\rho [i \mapsto v_1'\ f_1']@i \left\langle x_1.\ Q\right\rangle$
    by Cor. 1.
16. Suppose $W'.k > 0$ WLOG.
17. Pick v_2' with
    $|W'| \models^\rho v_1' \preceq^{\mathcal{V}} v_2' : \tau_1 \to \tau_2$,
    $\eta' = (\emptyset, \{\emptyset; [j \mapsto K[v_2'\ f_2']]\})$,
    $|W'| \overset{\text{rely}}{\sqsupseteq} U$.
18. Write $Q'' = \exists x_2''.\ x_1'' \preceq^{\mathcal{V}} x_2'' : \tau_1 \wedge j \rightarrowtail_{\mathrm{S}} K[v_2'\ x_2'']$.
19. $|W'|, \eta' \models^\rho [i \mapsto f_1']@i \left\langle x_1''.\ Q''\right\rangle$. by assumption and Lem. 17.
20. $|W'|, \eta' \models^\rho [i \mapsto f_1']@i \left\langle x_1''.\ Q'' \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W'|\right\rangle$ by Cor. 2.
21. Let $v_1'' \in \mathrm{Val}$.
22. Let W'', η'' with $W'', \eta'' \models^\rho Q''[v_1''/x_1''] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W'|$.

23. Suffices to show $W'', \eta'' \models^\rho [i \mapsto v_1'\ v_1'']@i \left\langle x_1.\ Q \right\rangle$ by Lem. 16.
24. Suffices to show $|W''|, \eta'' \models^\rho [i \mapsto v_1'\ v_1'']@i \left\langle x_1.\ Q \right\rangle$ by Cor. 1.
25. Suppose $W''.k > 0$ WLOG.
26. Pick v_2'' with
    $|W''| \models^\rho v_1'' \preceq^{\mathcal{V}} v_2'' : \tau_1$,
    $\eta'' = (\emptyset, \{\emptyset; [j \mapsto K[v_2'\ v_2'']]\})$,
    $|W''| \overset{\text{rely}}{\sqsupseteq} |W'|$.
27. Let $W''' \overset{\text{rely}}{\sqsupseteq} |W''|, \eta_f \# \eta''$.
28. Suppose $W'''.k > 0$ and $h, \Sigma : W''', \eta'' \otimes \eta_f$.
29. Write $v_1' = \textsf{rec}\ f(x).g_1'$ and $v_2' = \textsf{rec}\ f(x).g_2'$.
30. $h; [i \mapsto v_1'\ v_1''] \to h; [i \mapsto g_1'[v_1'/f, v_1''/x]]$.
31. Write $\Sigma = \{\emptyset; [j \mapsto K[v_2'\ v_2'']]\} \otimes \Sigma_0$.
32. Write $\Sigma' = \{\emptyset; [j \mapsto K[g_2'[v_2'/f, v_2''/x]]]\} \otimes \Sigma_0$.
33. $\Sigma \rightrightarrows \Sigma'$.
34. Write $\eta''' = (\emptyset, \{\emptyset; [j \mapsto K[g_2'[v_2'/f, v_2''/x]]]\})$.
35. $h, \Sigma : \triangleright W''', \eta'' \otimes \eta_f$ by Lem. 12.
36. $h, \Sigma' : \triangleright W''', \eta''' \otimes \eta_f$.
37. Suffices to show $\triangleright\, W''', \eta''' \models^\rho [i \mapsto g_1'[v_1'/f, v_1''/x]]@i \left\langle x_1.\ Q \right\rangle$.
38. Suffices to show $|\triangleright\, W'''|, \eta''' \models^\rho [i \mapsto g_1'[v_1'/f, v_1''/x]]@i \left\langle x_1.\ Q \right\rangle$ by Cor. 1.
39. Suppose $W'''.k > 0$ WLOG.
40. $|W'''| \models^\rho v_1' \preceq^{\mathcal{V}} v_2' : \tau_1 \to \tau_2$.
41. $|\triangleright\, W'''| = \triangleright |W'''|$.
42. $|\triangleright\, W'''| \models^\rho v_1'' \preceq^{\mathcal{V}} v_2'' : \tau_1$.
43. $|\triangleright\, W'''|, \eta''' \models^\rho [i \mapsto g_1'[v_1'/f, v_1''/x]]@i \left\langle x_1.\ Q \right\rangle$ by Lem. 20.

□

Lemma 22.

$$\frac{\Delta; \Gamma, f : \tau_1 \to \tau_2, x : \tau_1 \models e_1 \preceq e_2 : \tau_2}{\Delta; \Gamma \models \textsf{rec}\ f(x).e_1 \preceq \textsf{rec}\ f(x).e_2 : \tau_1 \to \tau_2}$$

Proof.

1. Let $U, \rho : \Delta \to \mathrm{VRel}, \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}$.
2. Suppose $\forall x \in \mathrm{dom}(\Gamma).\ U \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
3. Write $e_1' = e_1[\gamma_1/\Gamma], e_2' = e_2[\gamma_2/\Gamma]$.
4. Let K, i, j.
5. Write $\eta = (\emptyset, \{\emptyset; [j \mapsto K[\textsf{rec}\ f(x).e_2']]\})$.
6. Write $Q = \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau_1 \to \tau_2 \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]$.
7. Suffices to show $U, \eta \models^\rho [i \mapsto \textsf{rec}\ f(x).e_1']@i \left\langle x_1.\ Q \right\rangle$. by Lem. 17.
8. Suffices to show $U, \eta \models^\rho Q[\textsf{rec}\ f(x).e_1'/x_1]$.

9. Suffices to show $U \models^\rho \textbf{rec}\ f(x).e_1' \preceq^{\mathcal{V}} \textbf{rec}\ f(x).e_2' : \tau_1 \to \tau_2$.
10. Suffices to show $\forall U' \overset{\text{rely}}{\sqsupseteq} U.\ U' \models^\rho \textbf{rec}\ f(x).e_1' \preceq^{\mathcal{V}} \textbf{rec}\ f(x).e_2' : \tau_1 \to \tau_2$.
11. Proceed by induction on $U'.k$.
12. Suppose $U'.k > 0$.
13. Let $w_1, w_2, U'' \overset{\text{rely}}{\sqsupseteq} \triangleright U'$ with $U'' \models^\rho w_1 \preceq^{\mathcal{V}} w_2 : \tau_1$.
14. Let K, j, i.
15. Write $\eta' = (\emptyset, \{\emptyset; [j \mapsto K[e_2'[\textbf{rec}\ f(x).e_2'/f, w_2/x]]]\})$.
16. Write $Q' = \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \tau_2 \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]$.
17. Suffices to show $U'', \eta' \models^\rho [i \mapsto e_1'[\textbf{rec}\ f(x).e_1'/f, w_1/x]]@i\ \langle x_1.\ Q' \rangle$ by Lem. 20.
18. $\forall x \in \mathrm{dom}(\Gamma).\ U'' \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
19. $U'' \models^\rho \textbf{rec}\ f(x).e_1' \preceq^{\mathcal{V}} \textbf{rec}\ f(x).e_2' : \tau_1 \to \tau_2$ by induction hypothesis.
20. $U'', \eta' \models^\rho [i \mapsto e_1'[\textbf{rec}\ f(x).e_1'/f, w_1/x]]@i\ \langle x_1.\ Q' \rangle$ by assumption and Lem. 17.

□

3.3.4 CAS

Lemma 23. For $U.k \neq 0$ we have that $U \models^\rho \ell_1 \preceq^{\mathcal{V}} \ell_2 : \textbf{ref}(\overline{\tau})$ implies the existence of an $i \in \mathrm{dom}(U.\omega)$ such that we have

$$\mathcal{I}[\![U.\omega(i)]\!]_U = \left\{([\ell_1 \mapsto \overline{v_1}], \{[\ell_2 \mapsto \overline{v_2}]; \emptyset\}) \mid \bigwedge \overline{\triangleright U \models^\rho v_1 \preceq^{\mathcal{V}} v_2 : \tau}\right\}.$$

Lemma 24. Assume that we have $U \models^\rho v_1 \preceq^{\mathcal{V}} v_2 : \sigma$ and $U \models^\rho w_1 \preceq^{\mathcal{V}} w_2 : \sigma$. If $U.k \neq 0$ and there are η, h and Σ such that $h, \Sigma : U, \eta$ holds, then we have that

$$v_1 = w_1 \iff v_2 = w_2.$$

Lemma 25.

$$\frac{\Delta; \Gamma \models e_1 \preceq e_2 : \textbf{ref}(\overline{\tau}) \qquad \tau_n = \sigma \qquad \Delta; \Gamma \models f_1 \preceq f_2 : \sigma \qquad \Delta; \Gamma \models g_1 \preceq g_2 : \sigma}{\Delta; \Gamma \models \mathsf{CAS}(e_1[n], f_1, g_1) \preceq \mathsf{CAS}(e_2[n], f_2, g_2) : \mathbf{B}}$$

Proof.

1. Let $U, \rho : \Delta \to \mathrm{VRel}, \gamma_1, \gamma_2 : \mathrm{dom}(\Gamma) \to \mathrm{Val}$.
2. Suppose $\forall x \in \mathrm{dom}(\Gamma).\ U \models^\rho \gamma_1(x) \preceq^{\mathcal{V}} \gamma_2(x) : \Gamma(x)$.
3. Write $e_1' = e_1[\gamma_1/\Gamma], e_2' = e_2[\gamma_2/\Gamma]$.
4. Write $f_1' = f_1[\gamma_1/\Gamma], f_2' = f_2[\gamma_2/\Gamma]$.
5. Write $g_1' = g_1[\gamma_1/\Gamma], g_2' = g_2[\gamma_2/\Gamma]$.
6. Let K, i, j.
7. Write $\eta = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{CAS}(e_2'[n], f_2', g_2')]]\})$.
8. Write $Q = \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \mathbf{B} \wedge j \rightarrowtail_{\mathrm{S}} K[x_2]$.
9. Suffices to show $U, \eta \models^\rho [i \mapsto \mathsf{CAS}(e_1'[n], f_1', g_1')]@i\ \langle x_1.\ Q \rangle$. by Lem. 17.
10. Write $Q' = \exists x_2'.\ x_1' \preceq^{\mathcal{V}} x_2' : \textbf{ref}(\overline{\tau}) \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{CAS}(x_2'[n], f_2', g_2')]$.
11. $U, \eta \models^\rho [i \mapsto e_1']@i\ \langle x_1'.\ Q' \rangle$. by assumption and Lem. 17.

12. $U, \eta \models^\rho [i \mapsto e_1']@i \left\langle x_1'.\ Q' \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U \right\rangle$. by Cor. 2.
13. Let $v_1' \in \text{Val}$.
14. Let W', η' with $W', \eta' \models^\rho Q'[v_1'/x_1'] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} U$.
15. Suffices to show $W', \eta' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], f_1', g_1')]@i \left\langle x_1.\ Q \right\rangle$ by Lem. 16.
16. Suffices to show $|W'|, \eta' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], f_1', g_1')]@i \left\langle x_1.\ Q \right\rangle$ by Cor. 1.
17. Suppose $W'.k > 0$ WLOG.
18. Pick v_2' with
$|W'| \models^\rho v_1' \preceq^{\mathcal{V}} v_2' : \mathbf{ref}(\overline{\tau})$,
$\eta' = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{CAS}(v_2'[n], f_2', g_2')]]\})$,
$|W'| \overset{\text{rely}}{\sqsupseteq} U$.
19. Write $Q'' = \exists x_2''.\ x_1'' \preceq^{\mathcal{V}} x_2'' : \sigma \wedge j \rightarrowtail_{\text{S}} K[\mathsf{CAS}(v_2'[n], x_2'', g_2')]$.
20. $|W'|, \eta' \models^\rho [i \mapsto f_1']@i \left\langle x_1''.\ Q'' \right\rangle$ by assumption and Lem. 17.
21. $|W'|, \eta' \models^\rho [i \mapsto f_1']@i \left\langle x_1''.\ Q'' \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W'| \right\rangle$ by Cor. 2.
22. Let $v_1'' \in \text{Val}$.
23. Let W'', η'' with $W'', \eta'' \models^\rho Q''[v_1''/x_1''] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W'|$.
24. Suffices to show $W'', \eta'' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], v_1'', g_1')]@i \left\langle x_1.\ Q \right\rangle$ by Lem. 16.
25. Suffices to show $|W''|, \eta'' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], v_1'', g_1')]@i \left\langle x_1.\ Q \right\rangle$ by Cor. 1.
26. Suppose $W''.k > 0$ WLOG.
27. Pick v_2'' with
$|W''| \models^\rho v_1'' \preceq^{\mathcal{V}} v_2'' : \sigma$,
$\eta'' = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{CAS}(v_2'[n], v_2'', g_2')]]\})$,
$|W''| \overset{\text{rely}}{\sqsupseteq} |W'|$.
28. Write $Q''' = \exists x_2'''.\ x_1''' \preceq^{\mathcal{V}} x_2''' : \sigma \wedge j \rightarrowtail_{\text{S}} K[\mathsf{CAS}(v_2'[n], v_2'', x_2''')]$.
29. $|W''|, \eta' \models^\rho [i \mapsto g_1']@i \left\langle x_1'''.\ Q''' \right\rangle$ by assumption and Lem. 17.
30. $|W''|, \eta' \models^\rho [i \mapsto g_1']@i \left\langle x_1'''.\ Q'' \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W'| \right\rangle$ by Cor. 2.
31. Let $v_1''' \in \text{Val}$.
32. Let W''', η''' with $W''', \eta''' \models^\rho Q'''[v_1'''/x_1'''] \wedge \cdot \overset{\text{rely}}{\sqsupseteq} |W''|$.
33. Suffices to show $W''', \eta''' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], v_1'', v_1''')]@i \left\langle x_1.\ Q \right\rangle$ by Lem. 16.
34. Suffices to show $|W'''|, \eta''' \models^\rho [i \mapsto \mathsf{CAS}(v_1'[n], v_1'', v''')]@i \left\langle x_1.\ Q \right\rangle$ by Cor. 1.
35. Suppose $W'''.k > 0$ WLOG.
36. Pick v_2''' with
$|W'''| \models^\rho v_1''' \preceq^{\mathcal{V}} v_2''' : \sigma$,
$\eta''' = (\emptyset, \{\emptyset; [j \mapsto K[\mathsf{CAS}(v_2'[n], v_2'', v_2''')]]\})$,
$|W'''| \overset{\text{rely}}{\sqsupseteq} |W''|$.
37. Let $W'''' \overset{\text{rely}}{\sqsupseteq} W''', \eta_f \# \eta'''$.

38. Suppose $W''''.k > 0$ and $h, \Sigma : W'''', \eta''' \otimes \eta_f$.
39. Suppose $h; [i \mapsto \mathsf{CAS}(v_1'[n], v_1'', v''')] \to h'; T'$.
40. Suppose $W''''.k > 1$ WLOG.
41. Write $v_1' = \ell_1$ and $v_2' = \ell_2$.
42. Pick $\overline{v}, \overline{w}, h_0, \Sigma_0$ with
    $\bigwedge \overline{\triangleright |W''''| \models^\rho v \preceq^{\mathcal{V}} w : \tau}$,
    $h = [\ell_1 \mapsto \overline{v}] \uplus h_0$,
    $\Sigma = \{[\ell_2 \mapsto \overline{w}]; \emptyset\} \otimes \{\emptyset; [j \mapsto K[\mathsf{CAS}(v_2'[n], v_2'', v_2''')]]\} \otimes \Sigma_0$
    by Lem. 23.
43. $\triangleright |W''''| \models^\rho v_n \preceq^{\mathcal{V}} w_n : \sigma$.
44. $\triangleright |W''''| \models^\rho v_1'' \preceq^{\mathcal{V}} v_2'' : \sigma$.
45. $h, \Sigma : \triangleright |W''''|, \eta''' \otimes \eta_f$ by Lem. 12.
46. $v_n = v_1'' \Leftrightarrow w_n = v_2''$ by Lem. 24.

Case $\boxed{v_n = v_1'' \wedge w_n = v_2''}$

47. Write $v_n^\dagger = v_1'''$ and $v_m^\dagger = v_m, m \neq n$.
48. $h' = h[\ell_1 \mapsto \overline{v^\dagger}]$.
49. $T' = [i \mapsto \mathbf{true}]$.
50. Write $w_n^\dagger = v_2'''$ and $w_m^\dagger = w_m, m \neq n$.
51. Write $\Sigma' = \{[\ell_2 \mapsto \overline{w^\dagger}]; \emptyset\} \otimes \{\emptyset; [j \mapsto K[\mathbf{true}]]\} \otimes \Sigma_0$.
52. $\Sigma \rightrightarrows \Sigma'$.
53. Write $\eta'''' = (\emptyset, \{\emptyset; [j \mapsto K[\mathbf{true}]]\})$.
54. $\bigwedge \overline{\triangleright |W''''| \models^\rho v^\dagger \preceq^{\mathcal{V}} w^\dagger : \tau}$.
55. $h', \Sigma' : W'''', \eta'''' \otimes \eta_f$.
56. $h', \Sigma' : \triangleright W'''', \eta'''' \otimes \eta_f$ by Lem. 12.
57. $\triangleright W'''', \eta'''' \models^\rho [i \mapsto \mathbf{true}]@i \langle x_1.\ Q \rangle$.

Case $\boxed{v_n \neq v_1'' \wedge w_n \neq v_2''}$

58. $h' = h$.
59. $T' = [i \mapsto \mathbf{false}]$.
60. Write $\Sigma' = \{[\ell_2 \mapsto \overline{w}]; \emptyset\} \otimes \{\emptyset; [j \mapsto K[\mathbf{false}]]\} \otimes \Sigma_0$.
61. $\Sigma \rightrightarrows \Sigma'$.
62. Write $\eta'''' = (\emptyset, \{\emptyset; [j \mapsto K[\mathbf{false}]]\})$.
63. $h, \Sigma' : W'''', \eta'''' \otimes \eta_f$.
64. $h, \Sigma' : \triangleright W'''', \eta'''' \otimes \eta_f$ by Lem. 12 .
65. $\triangleright W'''', \eta'''' \models^\rho [i \mapsto \mathbf{false}]@i \langle x_1.\ Q \rangle$.

□

3.4 May-refinement

Theorem 1 (May-refinement)**.** Suppose $\cdot;\cdot \models e_1 \preceq e_2 : \mathbf{N}$ holds. Let h_1, h_2, i, j and n be arbitrary. If we have

$$\exists h_1', T_1.\ h_1; [i \mapsto e_1] \to^* h_1'; [i \mapsto n] \uplus T_1$$

then we also have

$$\exists h_2', T_2.\ h_2; [j \mapsto e_2] \to^* h_2'; [j \mapsto n] \uplus T_2.$$

Proof. Let M be the number of steps in the assumed reduction. Write

$$h_1; [i \mapsto e_1] = h_1^0; T_1^0 \to h_1^1; T_1^1 \to \cdots \to h_1^M; T_1^M = h_1'; [i \mapsto n] \uplus T_1.$$

We proceed to prove by induction the claim that for all $0 \leq m \leq M$ there are W_m, $\eta_m \# \eta$, and Σ_m with the following properties, where $\eta = (h_1, \{h_2; \emptyset\})$ and $\Sigma = \{h_2; [j \mapsto e_2]\}$:

- $W_m, \eta_m \models^\rho T_1^m @ i \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \mathbf{N} \wedge j \rightarrowtail_{\mathrm{S}} x_2 \right\rangle$.
- $h_1^m, \Sigma_m : W_m, \eta_m \otimes \eta$.
- $W_m.k = 1 + M - m$.
- $\Sigma \rightrightarrows \Sigma_m$.

Let us initially consider the base case $m = 0$. We choose $W_0 = (1 + M, \emptyset)$, $\eta_0 = (\emptyset, \{\emptyset; [j \mapsto e_2]\})$ and $\Sigma_0 = \{h_2; [j \mapsto e_2]\}$. The different properties are easily verified; the only nontrivial is the first and that follows from the initial assumption $\cdot;\cdot \models e_1 \preceq e_2 : \mathbf{N}$. The induction step comes down to unrolling the definition of threadpool triples; we omit the details.

Instantiating our claim at $m = M$ now gives us W_M, $\eta_M \# \eta$, and Σ_M such that:

- $W_M, \eta_M \models^\rho [i \mapsto n] \uplus T_1 @ i \left\langle x_1.\ \exists x_2.\ x_1 \preceq^{\mathcal{V}} x_2 : \mathbf{N} \wedge j \rightarrowtail_{\mathrm{S}} x_2 \right\rangle$.
- $h_1', \Sigma_M : W_M, \eta_M \otimes \eta$.
- $W_M = 1$.
- $\Sigma \rightrightarrows \Sigma_M$.

A final unrolling of the definition of threadpool triples and a few calculations gives us Σ' with $\Sigma \rightrightarrows \Sigma'$ and $\Sigma' = \{\emptyset; [j \mapsto n]\} \otimes \Sigma_0$. All that remains is to pick an element from the nonempty set Σ'. $\square$

4 Proof theory

NOTE: This section contains a **sketch** of an eventual proof theory for the logic. While the examples presented in the paper can be proved directly in the model, formulating certain lemmas (here presented as proof rules) facilitates higher-level reasoning.

4.1 Hypothetical reasoning

We give inference rules in hypothetical, natural deduction style, using $\mathcal{P}$ to record hypotheses:

$$\mathcal{P} \quad ::= \quad \cdot \mid \mathcal{P}, P$$

with the semantic interpretations

$$\begin{array}{rcl} W, \eta \models^\rho \mathcal{P} & \triangleq & \forall P \in \mathcal{P}.\ W, \eta \models^\rho P \\ \mathcal{P} \models Q & \triangleq & \forall W, \eta, \rho, \gamma.\ W, \eta \models^\rho \gamma\mathcal{P} \implies W, \eta \models^\rho \gamma Q \end{array}$$

where γ ranges over variable-to-value substitutions and ρ and γ are constrained to close both $\mathcal{P}$ and Q.

Lemma 26 (Inference weakening)**.** If $\cdot \models (P_1 \wedge \cdots \wedge P_n) \Rightarrow Q$ then for all $\mathcal{P}$ we have

$$\frac{\mathcal{P} \models P_1 \qquad \ldots \qquad \mathcal{P} \models P_n}{\mathcal{P} \models Q}$$

Proof.

1. Let $W, \eta \models^\rho \gamma\mathcal{P}$
2. $W, \eta \models^\rho \gamma P_i,\ 1 \leq i \leq n$ — by assumption
3. $W, \eta \models^\rho (\gamma P_1 \wedge \cdots \wedge \gamma P_n)$ — by defn of $\wedge$, induction on n
4. $W, \eta \models^\rho (\gamma P_1 \wedge \cdots \wedge \gamma P_n) \Rightarrow \gamma Q$ — by assumption
5. $W, \eta \models^\rho \gamma Q$ — by defn of $\Rightarrow$

□

We will use the above lemma extensively in proving the soundness of inference rules. When proving a schematic inference rule that is closed under variable substitution, we will freely prove the inference rule as if all assertions were closed (wrt variables), without loss of generality.

4.2 Laws of intuitionistic first-order logic

$$\frac{P \in \mathcal{P}}{\mathcal{P} \vdash P}\ \textsc{Asm} \qquad \frac{\mathcal{P} \vdash P(v) \qquad \mathcal{P} \vdash v = v'}{\mathcal{P} \vdash P(v')}\ \textsc{Eq} \qquad \frac{\mathcal{P} \vdash P \qquad \mathcal{P} \vdash Q}{\mathcal{P} \vdash P \wedge Q}\ \wedge\text{I} \qquad \frac{\mathcal{P} \vdash P \wedge Q}{\mathcal{P} \vdash P}\ \wedge\text{EL}$$

$$\frac{\mathcal{P} \vdash P \wedge Q}{\mathcal{P} \vdash Q}\ \wedge\text{ER} \qquad \frac{\mathcal{P} \vdash P \vee Q \qquad \mathcal{P}, P \vdash R \qquad \mathcal{P}, Q \vdash R}{\mathcal{P} \vdash R}\ \vee\text{E} \qquad \frac{\mathcal{P} \vdash P}{\mathcal{P} \vdash P \vee Q}\ \vee\text{IL}$$

$$\frac{\mathcal{P} \vdash Q}{\mathcal{P} \vdash P \vee Q}\ \vee\text{IR} \qquad \frac{\mathcal{P}, P \vdash Q}{\mathcal{P} \vdash P \Rightarrow Q}\ \Rightarrow\text{I} \qquad \frac{\mathcal{P} \vdash P \Rightarrow Q \qquad \mathcal{P} \vdash P}{\mathcal{P} \vdash Q}\ \Rightarrow\text{E}$$

$$\frac{\mathcal{P} \vdash P(y) \qquad y \text{ fresh for } \mathcal{P}, P(x)}{\mathcal{P} \vdash \forall x.P(x)}\ \forall\text{I} \qquad \frac{\mathcal{P} \vdash \forall x.P(x)}{\mathcal{P} \vdash P(v)}\ \forall\text{E}$$

$$\frac{\mathcal{P} \vdash \exists x.P(x) \qquad \mathcal{P}, P(y) \vdash Q \qquad y \text{ fresh for } \mathcal{P}, Q, P(x)}{\mathcal{P} \vdash Q}\ \exists\text{E} \qquad \frac{\mathcal{P} \vdash P(v)}{\mathcal{P} \vdash \exists x.P(x)}\ \exists\text{I}$$

4.3 Additional rules from BI

$$\begin{array}{rcl} P * Q & \iff & Q * P \\ (P * Q) * R & \iff & P * (Q * R) \\ P * \mathsf{emp} & \iff & P \\ (P \lor Q) * R & \iff & (P * R) \lor (Q * R) \\ (P \land Q) * R & \Rightarrow & (P * R) \land (Q * R) \\ (\exists x.\, P) * Q & \iff & \exists x.\, (P * Q) \\ (\forall x.\, P) * Q & \Rightarrow & \forall x.\, (P * Q) \end{array}$$

$$\frac{\mathcal{P}, P_1 \vdash Q_1 \qquad \mathcal{P}, P_2 \vdash Q_2}{\mathcal{P}, P_1 * P_2 \vdash Q_1 * Q_2}$$

4.4 The "later" modality

$$\frac{\mathcal{P} \vdash P}{\mathcal{P} \vdash \triangleright P}\ \textsc{Mono} \qquad \frac{\mathcal{P}, \triangleright P \vdash P}{\mathcal{P} \vdash P}\ \textsc{Löb}$$

$$\begin{array}{rcl} \triangleright(P \land Q) & \iff & \triangleright P \land \triangleright Q \\ \triangleright(P \lor Q) & \iff & \triangleright P \lor \triangleright Q \\ \triangleright(P \Rightarrow Q) & \iff & \triangleright P \Rightarrow \triangleright Q \\ \triangleright \forall x. P & \iff & \forall x. \triangleright P \\ \triangleright \exists x. P & \iff & \exists x. \triangleright P \\ \triangleright(P * Q) & \iff & \triangleright P * \triangleright Q \end{array}$$

4.5 The "reachably" modality and speculation

$$\frac{\mathcal{P} \vdash P \oplus Q}{\mathcal{P} \vdash \diamond P} \qquad \frac{\mathcal{P} \vdash P \oplus Q}{\mathcal{P} \vdash \diamond Q} \qquad \frac{\mathcal{P} \vdash \diamond P \qquad \mathcal{P} \vdash \diamond Q}{\mathcal{P} \vdash \diamond(P \oplus Q)}$$

$$\begin{array}{rcl} \diamond(P \oplus Q) & \iff & \diamond P \oplus \diamond Q \\ P * (Q \oplus R) & \iff & (P * Q) \oplus (P * R) \end{array}$$

4.6 Atomic Hoare logic

$$(\!|\mathsf{emp}|\!)\ \mathbf{new}\ \overline{v}\ (\!|\mathsf{ret.\ ret} \mapsto_{\mathrm{I}} (\overline{v})|\!)\ \textsc{Alloc} \qquad (\!|v \mapsto_{\mathrm{I}} (\overline{v})|\!)\ v[i]\ (\!|\mathsf{ret.\ ret} = v_i \land v \mapsto_{\mathrm{I}} (\overline{v})|\!)\ \textsc{Deref}$$

$$(\!|v \mapsto_{\mathrm{I}} (v_1, \ldots, v_n)|\!)\ v[i]\ :=\ v_i'\ (\!|\mathsf{ret.\ ret} = () \land v \mapsto_{\mathrm{I}} (v_1, \ldots, v_{i-1}, v_i', v_{i+1}, \ldots, v_n)|\!)\ \textsc{Assign}$$

$$(\!|v \mapsto_{\mathrm{I}} (v_1, \ldots, v_n)|\!)\ \mathsf{CAS}(v[i], v_i, v_i')\ (\!|\mathsf{ret.\ ret} = \mathbf{true} \land v \mapsto_{\mathrm{I}} (v_1, \ldots, v_{i-1}, v_i', v_{i+1}, \ldots, v_n)|\!)\ \textsc{CASTrue}$$

$$(\!|v \mapsto_{\mathrm{I}} (\overline{v}) \land v_o \neq v_i|\!)\ \mathsf{CAS}(v[i], v_o, v_i')\ (\!|\mathsf{ret.\ ret} = \mathbf{false} \land v \mapsto_{\mathrm{I}} (\overline{v})|\!)\ \textsc{CASFalse}$$

$$(\!|\mathsf{emp}|\!)\ \mathbf{inj}_i\ v\ (\!|\mathsf{ret.\ ret} \mapsto_{\mathrm{I}} \mathbf{inj}_i\ v|\!)\ \textsc{Inject} \qquad \frac{(\!|P|\!)\ a\ (\!|x.\ \diamond Q|\!)}{(\!|P|\!)\ a\ (\!|x.\ Q|\!)}\ \textsc{ASpecExec}$$

$$\frac{P \vdash P' \qquad (\!|P'|\!)\ a\ (\!|x.\ Q'|\!) \qquad Q' \vdash Q}{(\!|P|\!)\ a\ (\!|x.\ Q|\!)}\ \textsc{AConseq} \qquad \frac{(\!|P|\!)\ a\ (\!|x.\ Q|\!)}{(\!|P * R|\!)\ a\ (\!|x.\ Q * R|\!)}\ \textsc{AFrame}$$

$$\frac{(\!|P_1|\!)\ a\ (\!|x.\ Q|\!) \qquad (\!|P_2|\!)\ a\ (\!|x.\ Q|\!)}{(\!|P_1 \lor P_2|\!)\ a\ (\!|x.\ Q|\!)}\ \textsc{ADisj}$$

4.7 Concurrent Hoare logic

$$\frac{\langle P\rangle\ e\ \langle x.\ Q\rangle \qquad \forall x.\ \langle Q\rangle\ K[x]\ \langle y.\ R\rangle}{\langle P\rangle\ K[e]\ \langle y.\ R\rangle}\ \textsc{Bind} \qquad \frac{e \overset{\text{pure}}{\hookrightarrow} e' \qquad \langle P\rangle\ e'\ \langle Q\rangle}{\langle \triangleright P\rangle\ e\ \langle Q\rangle}\ \textsc{PureRed}_{\text{I}}$$

$$\frac{}{\langle \mathsf{emp}\rangle\ v\ \langle x.\ x = v \wedge \mathsf{emp}\rangle}\ \textsc{Return} \qquad \frac{P \vdash P' \qquad \langle P'\rangle\ e\ \langle x.\ Q'\rangle \qquad Q' \vdash Q}{\langle P\rangle\ e\ \langle x.\ Q\rangle}\ \textsc{Conseq}$$

$$\frac{\langle P\rangle\ e\ \langle x.\ Q\rangle}{\langle P * R\rangle\ e\ \langle x.\ Q * R\rangle}\ \textsc{Frame} \qquad \frac{P \text{ pure} \qquad \mathcal{P} \vdash P \qquad \langle P \wedge Q\rangle\ e\ \langle x.\ R\rangle}{\mathcal{P} \vdash \langle Q\rangle\ e\ \langle x.\ R\rangle}$$

$$\frac{\langle P_1\rangle\ e\ \langle x.\ Q\rangle \qquad \langle P_2\rangle\ e\ \langle x.\ Q\rangle}{\langle P_1 \vee P_2\rangle\ e\ \langle x.\ Q\rangle}\ \textsc{Disj} \qquad \frac{\langle P\rangle\ e\ \langle x.\ \diamond Q\rangle}{\langle P\rangle\ e\ \langle x.\ Q\rangle}\ \textsc{SpecExec}$$

$$\frac{\langle P\rangle\ e\ \langle x.\ Q * \triangleright \iota.I(\iota.s)\rangle \qquad \iota.I(\iota.s) \text{ token-pure}}{\langle \triangleright P\rangle\ a\ \langle x.\ Q \wedge \iota\rangle}\ \textsc{IslNew} \qquad \frac{(\!|P|\!)\ a\ (\!|x.\ Q|\!)}{\langle \triangleright P\rangle\ a\ \langle x.\ Q\rangle}\ \textsc{Private}$$

$$\frac{\begin{array}{c}P, Q \text{ token-pure} \\ \forall \iota \overset{\text{rely}}{\sqsupseteq} \iota_0.\ \exists \iota' \overset{\text{guar}}{\sqsupseteq} \iota.\ \exists Q.\ \ \iota', Q \vdash R \ \wedge\ (\!|\iota.I(\iota.s) * P|\!)\ a\ (\!|x.\ \triangleright\, \iota'.I(\iota'.s) * Q|\!)\end{array}}{\langle \iota_0 \wedge \triangleright P\rangle\ a\ \langle x.\ R\rangle}\ \textsc{IslFocus}$$

$$\frac{\begin{array}{c}\mathcal{P} \vdash \langle P\rangle\ e\ \langle x.\ \exists y.\ \triangleright\,(x \mapsto_{\text{I}} \mathsf{inj}_1\ y * Q_1) \vee \triangleright (x \mapsto_{\text{I}} \mathsf{inj}_2\ y * Q_2)\rangle \\ \mathcal{P} \vdash \forall x.\ \langle x \mapsto_{\text{I}} \mathsf{inj}_1\ y * Q_1\rangle\ e_1\ \langle \mathsf{ret}.\ R\rangle \qquad \mathcal{P} \vdash \forall x.\ \langle x \mapsto_{\text{I}} \mathsf{inj}_2\ y * Q_2\rangle\ e_2\ \langle \mathsf{ret}.\ R\rangle\end{array}}{\mathcal{P} \vdash \langle P\rangle\ \mathsf{case}(e, \mathsf{inj}_1\ y \Rightarrow e_1, \mathsf{inj}_2\ y \Rightarrow e_2)\ \langle \mathsf{ret}.\ R\rangle}$$

Note: the rule of conjunction

$$\frac{\langle P_1\rangle\ e\ \langle x.\ Q_1\rangle \qquad \langle P_2\rangle\ e\ \langle x.\ Q_2\rangle}{\langle P_1 \wedge P_2\rangle\ e\ \langle x.\ Q_1 \wedge Q_2\rangle}$$

is *probably unsound* in our logic (as it is with many concurrent separation logics), because, in the termination branch of the triple, the splitting of resources between the main thread and remaining threads can be chosen arbitrarily.

4.8 Refinement reasoning

$$\frac{\mathcal{P} \vdash v_1 \preceq^{\mathcal{V}} v_2 : \tau}{\mathcal{P} \vdash v_1 \preceq^{\mathcal{E}} v_2 : \tau} \qquad \mathcal{P} \vdash () \preceq^{\mathcal{V}} () : \mathbf{1} \qquad \mathcal{P} \vdash \mathsf{true} \preceq^{\mathcal{V}} \mathsf{true} : \mathbf{B} \qquad \mathcal{P} \vdash \mathsf{false} \preceq^{\mathcal{V}} \mathsf{false} : \mathbf{B}$$

$$\mathcal{P} \vdash n \preceq^{\mathcal{V}} n : \mathbf{N}$$

$$\frac{v_{\text{I}} = \mathsf{rec}\ f(x_{\text{I}}).e_{\text{I}} \qquad v_{\text{S}} = \mathsf{rec}\ f(x_{\text{S}}).e_{\text{S}} \qquad \mathcal{P}, x_{\text{I}} \preceq^{\mathcal{V}} x_{\text{S}} : \tau \vdash e_{\text{I}}[v_{\text{I}}/f] \preceq^{\mathcal{E}} e_{\text{S}}[v_{\text{S}}/f] : \tau'}{\triangleright\mathcal{P} \vdash v_{\text{I}} \preceq^{\mathcal{V}} v_{\text{S}} : \tau \to \tau'}$$

4.9 Additional rules

$$\frac{\triangleright\mathcal{P} \vdash P}{\mathcal{P} \vdash P} \qquad \mathcal{P} \vdash \langle \mathsf{emp} \rangle\ \mathsf{cons}(x, y)\ \langle z.\ z \mapsto_{\mathrm{I}} (x, y) \rangle \qquad \frac{\mathcal{P} \vdash \langle P \rangle\ e\ \langle x.\ \triangleright P' \rangle \qquad \mathcal{P} \vdash \forall x.\ \langle P' \rangle\ e'\ \langle y.\ P'' \rangle}{\mathcal{P} \vdash \langle P \rangle\ \mathbf{let}\ x = e\ \mathbf{in}\ e'\ \langle y.\ P'' \rangle}$$

$$\frac{\forall j, K.\ \mathcal{P} \vdash \left\langle x_{\mathrm{I}} \preceq^{\mathcal{V}} x_{\mathrm{S}} : \tau \wedge j \rightarrowtail_{\mathrm{S}} K[e_{\mathrm{S}}] \right\rangle\ e_{\mathrm{I}}\ \left\langle \mathsf{ret}_{\mathrm{I}}.\ \exists \mathsf{ret}_{\mathrm{S}}.\ \mathsf{ret}_{\mathrm{I}} \preceq^{\mathcal{V}} \mathsf{ret}_{\mathrm{S}} : \tau' \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{ret}_{\mathrm{S}}] \right\rangle}{\mathcal{P} \vdash \lambda x_{\mathrm{I}}.e_{\mathrm{I}} \preceq^{\mathcal{V}} \lambda x_{\mathrm{S}}.e_{\mathrm{S}} : \tau \to \tau'}$$

HoareRec

$$\frac{\mathcal{P}, \forall x.\ \langle P \rangle\ f\ x\ \langle \mathsf{ret}.\ Q \rangle \vdash \forall x.\ \langle P \rangle\ e\ \langle \mathsf{ret}.\ Q \rangle}{\triangleright\mathcal{P} \vdash \forall x.\ \langle P \rangle\ (\mathbf{rec}\ f(x).e)\ x\ \langle \mathsf{ret}.\ Q \rangle}$$

UnfoldRec

$$\frac{\begin{array}{c} v_{\mathrm{I}} = \mathbf{rec}\ f_{\mathrm{I}}(x_{\mathrm{I}}).e_{\mathrm{I}} \qquad v_{\mathrm{S}} = \mathbf{rec}\ f_{\mathrm{S}}(x_{\mathrm{S}}).e_{\mathrm{S}} \\ \mathcal{P}, \triangleright\forall x_{\mathrm{I}} \preceq^{\mathcal{V}} x_{\mathrm{S}} : \tau.\ e_{\mathrm{I}}[v_{\mathrm{I}}/f_{\mathrm{I}}] \preceq^{\mathcal{E}} e_{\mathrm{S}}[v_{\mathrm{S}}/f_{\mathrm{S}}] : \tau' \vdash \forall x_{\mathrm{I}} \preceq^{\mathcal{V}} x_{\mathrm{S}} : \tau.\ e_{\mathrm{I}}[v_{\mathrm{I}}/f_{\mathrm{I}}] \preceq^{\mathcal{E}} e_{\mathrm{S}}[v_{\mathrm{S}}/f_{\mathrm{S}}] : \tau' \end{array}}{\mathcal{P} \vdash v_{\mathrm{I}} \preceq^{\mathcal{V}} v_{\mathrm{S}} : \tau \to \tau'}$$

$$\frac{\begin{array}{c} \mathcal{P} \vdash \langle P \rangle\ e\ \langle x.\ \triangleright Q \wedge (x = \mathbf{true} \vee x = \mathbf{false}) \rangle \\ \mathcal{P} \vdash \langle Q[\mathbf{true}/x] \rangle\ e_{\mathbf{true}}\ \langle \mathsf{ret}.\ R \rangle \qquad \mathcal{P} \vdash \langle Q[\mathbf{false}/x] \rangle\ e_{\mathbf{false}}\ \langle \mathsf{ret}.\ R \rangle \end{array}}{\mathcal{P} \vdash \langle P \rangle\ \mathbf{if}\ e\ \mathbf{then}\ e_{\mathbf{true}}\ \mathbf{else}\ e_{\mathbf{false}}\ \langle \mathsf{ret}.\ R \rangle}$$

$$\frac{\begin{array}{c} \mathcal{P} \vdash \langle P \rangle\ e\ \langle x.\ \triangleright (x = () \wedge Q_1) \vee (\exists \ell.\ x = \ell \wedge Q_2) \rangle \\ \mathcal{P} \vdash \langle Q_1 \rangle\ e_1\ \langle \mathsf{ret}.\ R \rangle \qquad \mathcal{P} \vdash \forall \ell.\ \langle Q_2[\ell/x] \rangle\ e_2\ \langle \mathsf{ret}.\ R \rangle \end{array}}{\mathcal{P} \vdash \langle P \rangle\ \mathbf{case}(e, \mathbf{null} \Rightarrow e_1, \mathbf{some}(x) \Rightarrow e_2)\ \langle \mathsf{ret}.\ R \rangle}$$

4.10 Soundness of the inference rules

Lemma 27.

$$\frac{\langle P \rangle\ e\ \langle x.\ Q * \triangleright\iota.I(\iota.s) \rangle \qquad \iota.I(\iota.s)\ \text{token-pure}}{\langle \triangleright P \rangle\ a\ \langle x.\ Q \wedge \iota \rangle}\ \textsc{IslNew}$$

We employ Lem. 26 and so prove a simple implication on variable-closed assertions. We then prove the result by a straightforward induction on the step index in the underlying threadpool simulation. Thus, the proof boils down to the following internal lemma: if

$$h, \Sigma : W, \eta \otimes \eta_F \quad \text{and} \quad W, \eta \models^{\rho} Q * \triangleright\iota.I(\iota.s) * T@\mathsf{none}\ \langle \mathbf{true} \rangle$$

then

$$\exists \eta', W' \overset{\text{guar}}{\sqsupseteq} W.\ h, \Sigma : W', \eta' \otimes \eta_F \quad \text{and} \quad W', \eta' \models^{\rho} Q * \iota * T@\mathsf{none}\ \langle \mathbf{true} \rangle t$$

Proof.

1. $W = W_1 \otimes W_2 \otimes W_3, \quad \eta = \eta_1 \otimes \eta_2 \otimes \eta_3,$
   $W_1, \eta_1 \models^{\rho} Q, \quad W_2, \eta_2 \models^{\rho} \triangleright\iota.I(\iota.s), \quad W_3, \eta_3 \models^{\rho} T@\mathsf{none}\ \langle \mathbf{true} \rangle$
2. Let $W' = (W.k, W.\omega \uplus [i \mapsto \mathcal{I}[\![\iota]\!]^{\rho}_{k}])$
3. $W' \overset{\text{guar}}{\sqsupseteq} W$
4. $\triangleright|W'|, \eta_2 \models^{\rho} \triangleright\iota.I(\iota.s)$
5. Let $\eta' = \eta_1 \otimes \eta_3$
6. $h, \Sigma : W', \eta' \otimes \eta_F$
7. $W_2 \otimes W_3, \eta_3 \models^{\rho} T@\mathsf{none}\ \langle \mathbf{true} \rangle$ by framing
8. $W', \eta' \models^{\rho} Q * \iota * T@\mathsf{none}\ \langle \mathbf{true} \rangle$

□

Lemma 28.

$$\frac{(\!|P|\!)\ a\ (\!|x.\ Q|\!)}{\langle \triangleright P\rangle\ a\ \langle x.\ Q\rangle}\ \textsc{Private}$$

We employ Lem. 26 and so prove a simple implication on variable-closed assertions.

Proof.

1. Fix $i, W_0, W, \eta, \eta_F, \rho$
2. Suppose $W_0, \eta \models^\rho P, \quad W \overset{\text{rely}}{\sqsupseteq} W_0, \quad W.k > 0,$
   $\eta_F \# \eta, \quad h, \Sigma : W, \eta \otimes \eta_F, \quad h; [i \mapsto a] \to h'; T$
3. $\exists \eta_W.\quad (h, \Sigma) = \eta \otimes \eta_F \otimes \eta_W, \quad \eta_W \in \mathcal{I}[\![W.\omega]\!]_W$
4. $\exists v.\quad h; a \hookrightarrow h'; v, \quad T = [i \mapsto v]$ — by inversion on operational semantics
5. Let $\eta'_F = \eta_F \otimes \eta_W$
6. $\exists \eta' \# \eta'_F.\quad h' = (\eta' \otimes \eta'_F).h, \quad (\eta \otimes \eta'_F).\Sigma \rightrightarrows (\eta' \otimes \eta'_F).\Sigma, \quad \triangleright W, \eta' \models^\rho Q[v/x]$
   by assumption
7. Let $\Sigma' = (\eta' \otimes \eta'_F).\Sigma$
8. $\triangleright W \overset{\text{rely}}{\sqsupseteq} W$
9. $h', \Sigma' : \triangleright W, \eta' \otimes \eta_F$

□

Lemma 29.

$$\frac{\begin{array}{c} P, Q \text{ token-pure} \\ \forall \iota \overset{\text{rely}}{\sqsupseteq} \iota_0.\ \exists \iota' \overset{\text{guar}}{\sqsupseteq} \iota.\ \exists Q.\ \ \iota', Q \vdash R \ \wedge\ (\!|\iota.I(\iota.s) * P|\!)\ a\ (\!|x.\ \triangleright\, \iota'.I(\iota'.s) * Q|\!) \end{array}}{\langle \iota_0 \wedge \triangleright P\rangle\ a\ \langle x.\ R\rangle}\ \textsc{IslFocus}$$

Proof.

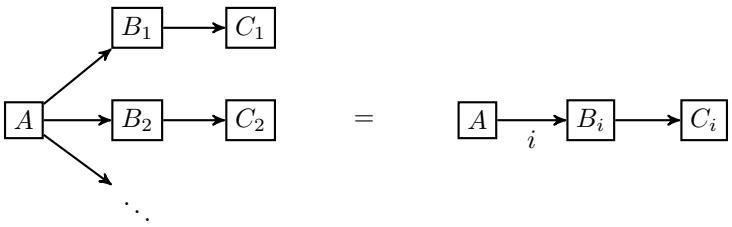
1. Fix i and $W_0, \eta \models^\rho \iota_0 \wedge P$
2. Suffices to show $W_0, \eta \models^\rho [i \mapsto e]@i\ \langle x.\ R\rangle$
3. Fix $W \overset{\text{rely}}{\sqsupseteq} W_0$ and $\eta_F \# \eta$
4. Suppose $W.k > 0$ and $h, \Sigma : W, \eta \otimes \eta_F$
5. $W, \eta \models^\rho \iota_0$
6. $\exists \iota \overset{\text{rely}}{\sqsupseteq} \iota_0, \omega_F, j.$
   $W.\omega = \omega_F \uplus [j \mapsto [\![\iota]\!]^\rho_{W.k}]$
7. $\exists \eta_\iota, \eta'_F.$ — by semantics of world satisfaction
   $(h, \Sigma) = \eta_\iota \otimes \eta \otimes \eta_F \otimes \eta'_F$
   $\eta_\iota \in \mathcal{I}[\![[\![\iota]\!]^\rho_{W.k}]\!]_W, \quad \eta'_F \in \mathcal{I}[\![\omega_F]\!]_W$
8. $\triangleright |W|, \eta_\iota \models \iota.J(\iota.s)$
9. $\exists \iota' \overset{\text{guar}}{\sqsupseteq} \iota,\ Q.$ — by assumption
   $\rho : (\iota' \wedge Q \Rightarrow R)$
   ρ: $(\!|\triangleright \iota.J(\iota.s) * P|\!)\ e\ (\!|x.\ \triangleright\, \iota'.J(\iota'.s) * Q|\!)$
10. Let $\widehat{\eta} = \eta \otimes \eta_\iota$
11. $|W|, \widehat{\eta} \models^\rho \triangleright \iota.J(\iota.s) * P$ — by (1, 8), token-purity of P

12. Let $\widehat{\eta}_F = \eta_F \otimes \eta'_F$
13. $h = (\widehat{\eta} \otimes \widehat{\eta}_F).h$
14. $h; e$ atomic — by (9)
15. Suppose $h; [i \mapsto e] \to h'; T$
16. Suppose $W.k > 1$ WLOG
17. $\exists v.\ h; e \hookrightarrow h'; v,\ T = [i \mapsto v]$ — by inversion, (14)
18. $\exists \widehat{\eta}'.\ h' = (\widehat{\eta}' \otimes \widehat{\eta}_F).h$, — by (9)
    $(\widehat{\eta} \otimes \widehat{\eta}_F).\Sigma \rightrightarrows (\widehat{\eta}' \otimes \widehat{\eta}_F).\Sigma$,
    $\triangleright|W|, \widehat{\eta}' \models^\rho \triangleright \iota'.J(\iota'.s) * Q[v/x]$
19. Let $W' = (W.k - 1, \lfloor \omega_F \rfloor_{W.k-1} \uplus [j \mapsto [\![\iota']\!]^\rho_{W.k-1}])$
20. $W' \overset{\text{guar}}{\sqsupseteq} W$
21. $\triangleright|W'| \overset{\text{rely}}{\sqsupseteq} |W'| \overset{\text{rely}}{\sqsupseteq} |\triangleright W| = \triangleright|W|$
22. $\exists \eta'_\iota, \eta'.\ \widehat{\eta}' = \eta' \otimes \eta'_\iota$, — by semantics of assertions, token-purity of Q
    $\triangleright|W'|, \eta'_\iota \models \iota'.J(\iota'.s)$,
    $W', \eta' \models^\rho Q[v/x]$
23. Write $\Sigma' = (\widehat{\eta}' \otimes \widehat{\eta}_F).\Sigma$
24. $\Sigma \rightrightarrows \Sigma'$
25. $h', \Sigma' : W', \eta' \otimes \eta_F$ — by (7, 22)
26. $W', \eta' \models^\rho Q[v/x] \wedge \iota'$
27. $W', \eta' \models^\rho R[v/x]$ — by (9)
28. $W', \eta' \models^\rho [i \mapsto v]@i \left\langle x.\ R \right\rangle$

□

5 Examples

We use a compact notation to draw structured branches:



5.1 Late/early choice

$$\begin{array}{rcl} \mathsf{rand} & \triangleq & \lambda().\ \textbf{let}\ y = \textbf{new false in}\ (\textbf{fork}\ y\ :=\ \textbf{true}); y[1] \\ \mathsf{lateChoice} & \triangleq & \lambda x.\ x\ :=\ 0; \mathsf{rand}() \\ \mathsf{earlyChoice} & \triangleq & \lambda x.\ \textbf{let}\ r = \mathsf{rand}()\ \textbf{in}\ x\ :=\ 0; r \end{array}$$

No internal protocol.

We use the *atomic Hoare triple* to show the details of the proof outline, by appeal to the IslFocus rule. Uses of the rules SpecExec and ASpecExec to rewrite a poscondition are marked with the notation $\rightrightarrows$ (written between the original and new postcondition).

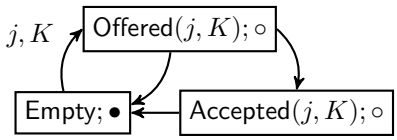
We have $\langle \mathsf{emp} \rangle\ \mathsf{rand}()\ \langle \mathsf{ret}.\ \mathsf{ret} = \textbf{true} \lor \mathsf{ret} = \textbf{false} \rangle$.

$\langle x_\mathrm{I} \preceq^{\mathcal{V}} x_\mathrm{S} : \textbf{ref}(\mathbf{N}) \land j \rightarrowtail_\mathrm{S} K[\mathsf{earlyChoice}(x_\mathrm{S})] \rangle$
$\quad (\!|(\exists y_\mathrm{I}, y_\mathrm{S}.\ y_\mathrm{I} \preceq^{\mathcal{V}} y_\mathrm{S} : \mathbf{N}\ \land\ (x_\mathrm{I} \mapsto_\mathrm{I} y_\mathrm{I}\ *\ x_\mathrm{S} \mapsto_\mathrm{S} y_\mathrm{S}))\ *\ j \rightarrowtail_\mathrm{S} K[\mathsf{earlyChoice}(x_\mathrm{S})]|\!)$
$\quad (\!|x_\mathrm{I} \mapsto_\mathrm{I} -\ *\ x_\mathrm{S} \mapsto_\mathrm{S} -\ *\ j \rightarrowtail_\mathrm{S} K[\mathsf{earlyChoice}(x_\mathrm{S})]|\!)$
$\quad\ x_\mathrm{I}\ :=\ 0$
$\quad (\!|x_\mathrm{I} \mapsto_\mathrm{I} 0\ *\ x_\mathrm{S} \mapsto_\mathrm{S} -\ *\ j \rightarrowtail_\mathrm{S} K[\mathsf{earlyChoice}(x_\mathrm{S})]|\!) \rightrightarrows$
$\quad (\!|x_\mathrm{I} \mapsto_\mathrm{I} 0\ *\ x_\mathrm{S} \mapsto_\mathrm{S} 0\ *\ (j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}])|\!)$
$\langle x_\mathrm{I} \preceq^{\mathcal{V}} x_\mathrm{S} : \textbf{ref}(\mathbf{N}) \land (j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}]) \rangle$
$\langle j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}] \rangle$
$\quad \mathsf{rand}()$
$\langle \mathsf{ret}.\ (\mathsf{ret} = \textbf{true} \lor \mathsf{ret} = \textbf{false}) * (j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}]) \rangle$
$\langle \mathsf{ret}.\ (\mathsf{ret} = \textbf{true} * (j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}])) \lor (\mathsf{ret} = \textbf{false} * (j \rightarrowtail_\mathrm{S} K[\textbf{true}] \oplus j \rightarrowtail_\mathrm{S} K[\textbf{false}])) \rangle \rightrightarrows$
$\langle \mathsf{ret}.\ j \rightarrowtail_\mathrm{S} K[\mathsf{ret}] \rangle$

5.2 Red/blue flags

$\mathsf{redFlag} \triangleq \lambda().\ \textbf{let}\ \mathit{flag} = \textbf{new}\ \textbf{true},\ \mathit{chan} = \textbf{new}\ 0\ \textbf{in}\ \{$
  $\mathsf{flip} = \textbf{rec}\ \mathsf{try}().$
    $\textbf{if}\ \mathsf{CAS}(\mathit{chan}, 1, 2)\ \textbf{then}\ ()\ \textbf{else}$
    $\textbf{if}\ \mathsf{CAS}(\mathit{flag}, \textbf{true}, \textbf{false})\ \textbf{then}\ ()\ \textbf{else}$
    $\textbf{if}\ \mathsf{CAS}(\mathit{flag}, \textbf{false}, \textbf{true})\ \textbf{then}\ ()\ \textbf{else}$
    $\textbf{if}\ \mathsf{CAS}(\mathit{chan}, 0, 1)\ \textbf{then}$
      $\textbf{if}\ \mathsf{CAS}(\mathit{chan}, 1, 0)\ \textbf{then}\ \mathsf{try}()\ \textbf{else}\ \mathit{chan} := 0$
    $\textbf{else}\ \mathsf{try}(),$
  $\mathsf{read} = \lambda().\ \mathit{flag}[1]$
$\}$
$\mathsf{blueFlag} \triangleq \lambda().\ \textbf{let}\ \mathit{flag} = \textbf{new}\ \textbf{true}, \mathsf{lock} = \textbf{new}\ \textbf{false}\ \textbf{in}\ \{$
  $\mathsf{flip} = \lambda().\ \mathsf{sync}(\mathsf{lock})\ \{\ \mathit{flag} := \textbf{not}\ \mathit{flag}[1]\ \},$
  $\mathsf{read} = \lambda().\ \mathsf{sync}(\mathsf{lock})\ \{\ \mathit{flag}[1]\ \}$
$\}$

An interesting aspect of this example: it does not matter which order we perform the *top-level* CASes in. Each CAS either succeeds and completes the spec, or fails and leaves our knowledge unchanged. The correctness proof is given at several levels of detail: the protocol, a high-level proof outline, an outline for each use of island focusing, and finally the (interesting) atomic triples. We begin with the protocol:



We use the shorthand $\exists x : \mathbf{B}.P$ for $\exists x.\ (x = \textbf{true} \lor x = \textbf{false}) \land P$ in defining the interpretation:

$$\begin{array}{rcl}
Q & \triangleq & \exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} x * \mathsf{lock} \mapsto_{\mathrm{S}} \textbf{false} \\
I(s) & \triangleq & Q * I_0(s) \\
I_0(\mathsf{Empty}) & \triangleq & \mathit{chan} \mapsto_{\mathrm{I}} 0 \\
I_0(\mathsf{Offered}(j, K)) & \triangleq & \mathit{chan} \mapsto_{\mathrm{I}} 1 * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] \\
I_0(\mathsf{Accepted}(j, K)) & \triangleq & \mathit{chan} \mapsto_{\mathrm{I}} 2 * j \rightarrowtail_{\mathrm{S}} K[()]
\end{array}$$

Let θ be the resulting transition system. Let $\mathsf{blueFlagBody}$ be the body of $\mathsf{blueFlag}$. Fix K, i.

$\langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{blueFlagBody}] \rangle$
  $\textbf{let}\ \mathit{flag}_{\mathrm{I}} = \textbf{new}\ \textbf{true}\ \textbf{in}$
$\langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{blueFlagBody}] * \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} \textbf{true} \rangle$
  $\textbf{let}\ \mathit{chan} = \textbf{new}\ 0\ \textbf{in}$
$\langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{blueFlagBody}] * \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} \textbf{true} * \mathit{chan} \mapsto_{\mathrm{I}} 0 \rangle \rightrightarrows$
$\langle j \rightarrowtail_{\mathrm{S}} K[\{\mathsf{flip} = \mathsf{flip}_{\mathrm{S}}, \mathsf{read} = \mathsf{read}_{\mathrm{S}}\}] * \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} \textbf{true} * \mathit{chan} \mapsto_{\mathrm{I}} 0 * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} \textbf{true} * \mathsf{lock} \mapsto_{\mathrm{S}} \textbf{false} \rangle$
$\langle j \rightarrowtail_{\mathrm{S}} K[\{\mathsf{flip} = \mathsf{flip}_{\mathrm{S}}, \mathsf{read} = \mathsf{read}_{\mathrm{S}}\}] * I(\mathsf{Empty}) \rangle$
$\langle j \rightarrowtail_{\mathrm{S}} K[\{\mathsf{flip} = \mathsf{flip}_{\mathrm{S}}, \mathsf{read} = \mathsf{read}_{\mathrm{S}}\}] \land (\theta, I, \mathsf{Empty}, \emptyset) \rangle$

Now we must show that $\mathsf{flip}_{\mathrm{I}}$ refines $\mathsf{flip}_{\mathrm{S}}$ (and similarly for read, which is trivial). We may assume the pure assertion $(\theta, I, \mathsf{Empty}, \emptyset)$. We then appeal to rule UnfoldRec (these details are elided, but standard).

Let $P = (\theta, I, \mathsf{Empty}, \emptyset) \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]$. First we give a high-level proof outline:

$\langle P \rangle$ **if** CAS($chan, 1, 2$) **then** $\langle j \rightarrowtail_{\mathrm{S}} K[()] \rangle$ () $\langle \mathsf{ret.\ ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
else $\langle P \rangle$ **if** CAS($flag$, **true**, **false**) **then** $\langle j \rightarrowtail_{\mathrm{S}} K[()] \rangle$ () $\langle \mathsf{ret.\ ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
else $\langle P \rangle$ **if** CAS($flag$, **false**, **true**) **then** $\langle j \rightarrowtail_{\mathrm{S}} K[()] \rangle$ () $\langle \mathsf{ret.\ ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
else $\langle P \rangle$ **if** CAS($chan, 0, 1$) **then** $\langle (\theta, I, \mathsf{Offered}(j, K), \bullet) \rangle$
  if CAS($chan, 1, 0$)
  then $\langle P \rangle$ try() $\langle \mathsf{ret.\ ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
  else $\langle (\theta, I, \mathsf{Accepted}(j, K), \bullet) \rangle$ $chan := 0$ $\langle \mathsf{ret.\ ret} = () \wedge (\theta, I, \mathsf{Empty}, \emptyset) \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
else $\langle P \rangle$ try() $\langle \mathsf{ret.\ ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$

For each of the CAS and assignment expressions, we apply the island focusing rule. The rule requires that for each rely-future state of the focused island, we can find a guarantee-future state and postcondition that together imply a common postcondition. Thus, proofs by island focusing generally require case analysis based on the island's state. For each use of island focusing, we give the overall (nonatomic) Hoare triple we are trying to prove, and then list underneath a series of atomic pre/post-conditions that together give a comprehensive case analysis of the rely-future states.

For the first CAS, the protocol may be in any state, but the CAS will only succeed if the state is Offered:

$\langle P \rangle$ CAS($chan, 1, 2$) $\langle \mathsf{ret.}\ (\mathsf{ret} = \textbf{true} * j \rightarrowtail_{\mathrm{S}} K[()]) \vee (\mathsf{ret} = \textbf{false} * P) \rangle$

$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Empty})$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Empty})$
$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Offered}(j', K'))$	$\mathsf{ret.\ ret} = \textbf{true} * j \rightarrowtail_{\mathrm{S}} K[()] \quad * I(\mathsf{Accepted}(j', K'))$
$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Accepted}(j', K'))$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Accepted}(j', K'))$

For the next pair of CASes, the protocol state is irrelevant to the success of the CAS, since we are attempting to perform the flip directly:

$\langle P \rangle$ CAS($flag$, **true**, **false**) $\langle \mathsf{ret.}\ (\mathsf{ret} = \textbf{true} * j \rightarrowtail_{\mathrm{S}} K[()]) \vee (\mathsf{ret} = \textbf{false} * P) \rangle$

$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(s)$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(s)$

$\langle P \rangle$ CAS($flag$, **false**, **true**) $\langle \mathsf{ret.}\ (\mathsf{ret} = \textbf{true} * j \rightarrowtail_{\mathrm{S}} K[()]) \vee (\mathsf{ret} = \textbf{false} * P) \rangle$

$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(s)$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(s)$

In the final top-level CAS, we attempt to make an offer, which succeeds only when the starting state is Empty. If successful, we transfer control of our specification resource:

$\langle P \rangle$ CAS($chan, 0, 1$) $\langle \mathsf{ret.}\ (\mathsf{ret} = \textbf{true} \wedge (\theta, I, \mathsf{Offered}(j, K), \bullet)) \vee (\mathsf{ret} = \textbf{false} \wedge P) \rangle$

$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Empty})$	$\mathsf{ret.\ ret} = \textbf{true} \quad * I(\mathsf{Offered}(j, K))$
$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Offered}(j', K'))$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Offered}(j', K'))$
$j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Accepted}(j', K'))$	$\mathsf{ret.\ ret} = \textbf{false} * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()] * I(\mathsf{Accepted}(j', K'))$

Once the offer is made, we attempt to withdraw it. Withdrawing succeeds only when the offer has not been accepted. Notice that, due to our ownership of the token, Empty is not a possible state.

$\langle (\theta, I, \mathsf{Offered}(j, K), \bullet) \rangle$ CAS($chan, 1, 0$) $\langle \mathsf{ret.}\ (\mathsf{ret} = \textbf{true} * P) \vee (\mathsf{ret} = \textbf{false} * (\theta, I, \mathsf{Accepted}(j, K), \bullet)) \rangle$

$I(\mathsf{Offered}(j, K))$	$\mathsf{ret.}\ (\mathsf{ret} = \textbf{true} \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]) * I(\mathsf{Empty})$
$I(\mathsf{Accepted}(j, K))$	$\mathsf{ret.\ ret} = \textbf{false} \quad * I(\mathsf{Accepted}(j', K'))$

If we did *not* succeed in withdrawing the offer, we can conclude that the state is at least Accepted. Due to our token ownership, that is the only state we need to consider when clearing the offer field:

$\langle (\theta, I, \mathsf{Accepted}(j, K), \bullet) \rangle$ $chan := 0$ $\langle \mathsf{ret.\ ret} = () * j \rightarrowtail_{\mathrm{S}} K[()] \rangle$

$I(\mathsf{Accepted}(j, K))$	$\mathsf{ret.\ ret} = () * j \rightarrowtail_{\mathrm{S}} K[()] * I(\mathsf{Empty})$

Finally, we give detailed derivations of the "interesting" atomic triples needed for the uses of island focusing above. Generally, the interesting cases are those where the CAS succeeds, or where nontrivial information about the protocol state is discovered.

The first top-level CAS succeeds in the Offered state:

$(\!|I(\mathsf{Offered}(j', K')) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathit{chan} \mapsto_{\mathrm{I}} 1 * Q * j' \rightarrowtail_{\mathrm{S}} K'[\mathsf{flip}_{\mathrm{S}}()] * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$\mathsf{CAS}(\mathit{chan}, 1, 2)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 2) * Q * j' \rightarrowtail_{\mathrm{S}} K'[\mathsf{flip}_{\mathrm{S}}()] * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 2) * \exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} x \quad * j' \rightarrowtail_{\mathrm{S}} K'[\mathsf{flip}_{\mathrm{S}}()] * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!) \rightrightarrows$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 2) * \exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} \neg x * j' \rightarrowtail_{\mathrm{S}} K'[()] * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!) \rightrightarrows$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 2) * \exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} x \quad * j' \rightarrowtail_{\mathrm{S}} K'[()] * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$(\!|\mathsf{ret}.\ \mathsf{ret} = \mathbf{true} * \mathit{chan} \mapsto_{\mathrm{I}} 2 * \exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} x \quad * j' \rightarrowtail_{\mathrm{S}} K'[()] * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$(\!|\mathsf{ret}.\ \mathsf{ret} = \mathbf{true} * I(\mathsf{Accepted}(j', K')) * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$

We prove the second CAS for any state s:
$(\!|I(s) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\exists x : \mathbf{B}.\ \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} x * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} x * I_0(s) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$\mathsf{CAS}(\mathit{flag}, \mathbf{true}, \mathbf{false})$
$(\!|\mathsf{ret}.\ ((\mathsf{ret} = \mathbf{true} * \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} \mathbf{false} * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} \mathbf{true}) \vee (\mathsf{ret} = \mathbf{false} * \mathit{flag}_{\mathrm{I}} \mapsto_{\mathrm{I}} \mathbf{false} * \mathit{flag}_{\mathrm{S}} \mapsto_{\mathrm{S}} \mathbf{false}))$
$\quad * I_0(s) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} * I(s) * j \rightarrowtail_{\mathrm{S}} K[()])$
$\quad \vee\ (\mathsf{ret} = \mathbf{false} * I(s) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()])|\!)$

The proof for $\mathsf{CAS}(\mathit{flag}, \mathbf{false}, \mathbf{true})$ is symmetric.

That leaves the final top-level CAS, in which we make an offer:
$(\!|I(\mathsf{Empty}) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathit{chan} \mapsto_{\mathrm{I}} 0 * Q * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$\mathsf{CAS}(\mathit{chan}, 0, 1)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 1) * Q * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathsf{ret}.\ \mathsf{ret} = \mathbf{true} * I(\mathsf{Offered}(j, K))|\!)$

We are now in a state where we own the token.
For the inner CAS, we therefore need to consider only two possible future states:
$(\!|I(\mathsf{Offered}(j, K))|\!)$
$(\!|\mathit{chan} \mapsto_{\mathrm{I}} 1 * Q * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$\mathsf{CAS}(\mathit{chan}, 1, 0)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{true} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 0) * Q * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$
$(\!|\mathsf{ret}.\ \mathsf{ret} = \mathbf{true} * I(\mathsf{Empty}) * j \rightarrowtail_{\mathrm{S}} K[\mathsf{flip}_{\mathrm{S}}()]|\!)$

$(\!|I(\mathsf{Accepted}(j, K))|\!)$
$(\!|\mathit{chan} \mapsto_{\mathrm{I}} 2 * Q * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$\mathsf{CAS}(\mathit{chan}, 1, 0)$
$(\!|\mathsf{ret}.\ (\mathsf{ret} = \mathbf{false} \wedge \mathit{chan} \mapsto_{\mathrm{I}} 2) * Q * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$(\!|\mathsf{ret}.\ \mathsf{ret} = \mathbf{false} * I(\mathsf{Accepted}(j, K))|\!)$

Finally, if the inner CAS fails, there is only one rely-future state: $\mathsf{Accepted}(j, K)$
Thus, we know exactly what the assignment to the channel will see:
$(\!|I(\mathsf{Accepted}(j, K))|\!)$
$(\!|Q * \mathit{chan} \mapsto_{\mathrm{I}} 2 * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$\mathit{chan} := 0$
$(\!|Q * \mathit{chan} \mapsto_{\mathrm{I}} 0 * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$
$(\!|I(\mathsf{Empty}) * j \rightarrowtail_{\mathrm{S}} K[()]|\!)$

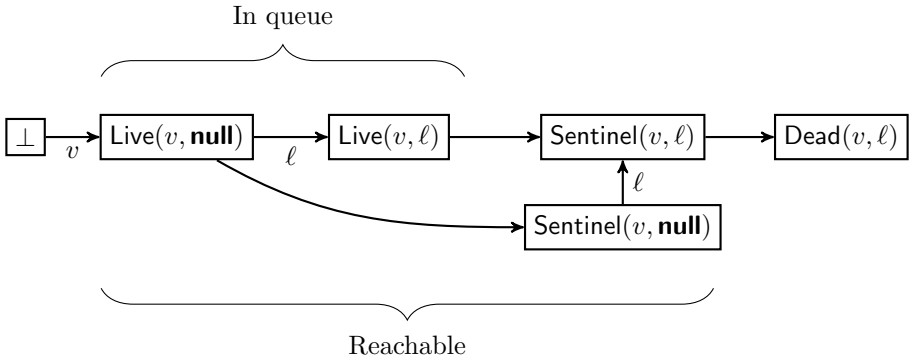
5.3 Michael-Scott queue

```
MSQ: ∀α.1 → { enq : α → 1, deq : 1 → ref?(α) }
MSQ ≜ Λ. λ(). let head = new cons(null, null) in {
  enq = λx. let n = cons(some(new x), null) in
            let rec try(c). case c[2] of
                null ⇒ if CAS(c[2], null, n) then () else try(c)
              | some(c') ⇒ try(c')
            in try(getVal(head[1])),
  deq = rec try().
    let c = head[1], n = getVal(c) in case n[2] of
        null ⇒ null        (* queue is empty *)
      | some(n') ⇒ if CAS(head[1], c, n') then n'[1] else try()
}
```

```
CGQ: ∀α.1 → { enq : α → 1, deq : 1 → ref?(α) }
CGQ ≜ Λ. λ(). let head = new null, lock = new false in {
  deq = λ(). sync(lock) { case head[1] of }
    null ⇒ null  |  some(n) ⇒ head[1] := n[2]; some(new n[1]),
  enq = λx. sync(lock) {  }
            case head[1] of
                null ⇒ head[1] := cons(x, null)
              | some(c) ⇒
                let rec try(c). case c[2] of
                    null ⇒ c[2] := cons(x, null)
                  | some(c') ⇒ try(c')
                in try(c)
}
```

Each instance of the MSQueue will have a fixed location for the head reference, and similarly on the specification side. We fix these as head_I and head_S respectively. In addition, the spec has a lock, lock, of type **ref**(**B**).

For every location ℓ, we have an instance of the following transition system, which tracks the life story of a node at that location:



This leads us to our state space—S_0 is per-location, while S gives the state space for the whole island:

$$
\begin{array}{rcl}
S_0 & \triangleq & \{\bot\} \\
 & \cup & \{\mathsf{Live}(v, v') \mid v, v' \in \mathrm{Val}\} \\
 & \cup & \{\mathsf{Sentinel}(v, v') \mid v, v' \in \mathrm{Val}\} \\
 & \cup & \{\mathsf{Dead}(v, \ell) \mid v \in \mathrm{Val}, \ell \in \mathrm{Loc}\} \\
S & \triangleq & \mathrm{Loc} \xrightarrow{\mathrm{fin}} S_0
\end{array}
$$

The notation $\stackrel{\text{fin}}{\rightharpoonup}$ here means that a state only maps finitely-many locations to a non-$\bot$ state—so $\bot$ is a useful pun.

We define $\rightsquigarrow_0$ according to the transition system drawn above, and lift this pointwise to $\rightsquigarrow$. There are no tokens in this example, so all that's left is the interpretation. We use a pattern-matching notation on states s of the space S; when nothing matches, the interpretation is ff.

$$\begin{array}{rcl} I(s) & \triangleq & \mathsf{head}_{\mathrm{I}} \mapsto_{\mathrm{I}} \ell_0 * \ell_0 \mapsto_{\mathrm{I}} (v_0, v_{\mathrm{I}}) * \mathsf{lock} \mapsto_{\mathrm{S}} \textbf{false} * \exists v_{\mathrm{S}}.\, \mathsf{head}_{\mathrm{S}} \mapsto_{\mathrm{S}} v_{\mathrm{S}} * \mathsf{link}(v_{\mathrm{I}}, v_{\mathrm{S}}, s_L) \\ & * & \ast_{\ell \in \mathrm{dom}(s_D)} \exists v, \ell'. s_D(\ell) = \mathsf{Dead}(v, \ell') * s(\ell') \neq \bot * \ell \mapsto_{\mathrm{I}} (v, \ell') \\ \multicolumn{3}{r}{\text{when } s = [\ell_0 \mapsto \mathsf{Sentinel}(v_0, v_{\mathrm{I}})] \uplus s_L \uplus s_D} \end{array}$$

$$\begin{array}{rcl} \mathsf{link}(\textbf{null}, \textbf{null}, \emptyset) & \triangleq & \mathsf{emp} \\ \mathsf{link}(\ell_{\mathrm{I}}, \ell_{\mathrm{S}}, [\ell_{\mathrm{I}} \mapsto \mathsf{Live}(v_{\mathrm{I}}, v'_{\mathrm{I}})] \uplus s) & \triangleq & \exists \ell, v_{\mathrm{S}}, v'_{\mathrm{S}}.\, \ell \mapsto_{\mathrm{I}} v_{\mathrm{I}} * v_{\mathrm{I}} \preceq^{\mathcal{V}} v_{\mathrm{S}} : \alpha \\ & * & \ell_{\mathrm{I}} \mapsto_{\mathrm{I}} (\ell, v'_{\mathrm{I}}) * \ell_{\mathrm{S}} \mapsto_{\mathrm{I}} (v_{\mathrm{S}}, v'_{\mathrm{S}}) * \mathsf{link}(v'_{\mathrm{I}}, v'_{\mathrm{S}}, s) \end{array}$$

Let θ be the transition system above. We use the shorthand

$$x \propto s_0 \quad \triangleq \quad (\theta, I, [x \mapsto s_0], \emptyset)$$

for $s_0 \in S_0$.

5.3.1 Proof outline for enq

Let

$$P \triangleq (\theta, I, \emptyset, \emptyset) * x_{\mathrm{I}} \preceq^{\mathcal{V}} x_{\mathrm{S}} : \alpha * j \rightarrowtail_{\mathrm{S}} K[\mathsf{enq}_{\mathrm{S}}(x_{\mathrm{S}})]$$

$\langle P \rangle$
  let $n = \mathsf{cons}(\textbf{some}(\textbf{new}\ x_{\mathrm{I}}), \textbf{null})$ **in**
$\langle P * \exists \ell.\ n \mapsto_{\mathrm{I}} (\ell, \textbf{null}) * \ell \mapsto_{\mathrm{I}} x_{\mathrm{I}} \rangle$
  let rec $\mathsf{try}(c)$.
  $\langle P * c \propto \mathsf{Live}(-,-) * \exists \ell.\ n \mapsto_{\mathrm{I}} (\ell, \textbf{null}) * \ell \mapsto_{\mathrm{I}} x_{\mathrm{I}} \rangle$
    case $c[2]$ **of**
      null $\Rightarrow$ $\langle P * c \propto \mathsf{Live}(-,-) * \exists \ell.\ n \mapsto_{\mathrm{I}} (\ell, \textbf{null}) * \ell \mapsto_{\mathrm{I}} x_{\mathrm{I}} \rangle$
        if $\mathsf{CAS}(c[2], \textbf{null}, n)$
        then $\langle n \propto \mathsf{Live}(x_{\mathrm{I}}, \textbf{null}) * j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
          $()$
        else $\langle P * c \propto \mathsf{Live}(-,-) * \exists \ell.\ n \mapsto_{\mathrm{I}} (\ell, \textbf{null}) * \ell \mapsto_{\mathrm{I}} x_{\mathrm{I}} \rangle$
          $\mathsf{try}(c)$
      | **some**(c') $\Rightarrow$ $\langle P * c' \propto \mathsf{Live}(-,-) * \exists \ell.\ n \mapsto_{\mathrm{I}} (\ell, \textbf{null}) * \ell \mapsto_{\mathrm{I}} x_{\mathrm{I}} \rangle$
        $\mathsf{try}(c')$
  $\langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$
  in $\mathsf{try}(\mathsf{getVal}(\mathsf{head}[1]))$
$\langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle$

At the atomic triple level, the reasoning depends on a case analysis of rely-future states, which determine in particular whether the CAS succeeds. Because Dead nodes must have non-**null** next pointers, the CAS will never succeed on a Dead node.

5.3.2 Proof outline for deq

Stable assumptions: $(\theta, I, \emptyset, \emptyset)$.
Let $P \triangleq j \rightarrowtail_{\mathrm{S}} K[\mathsf{deq}_{\mathrm{S}}()]$.

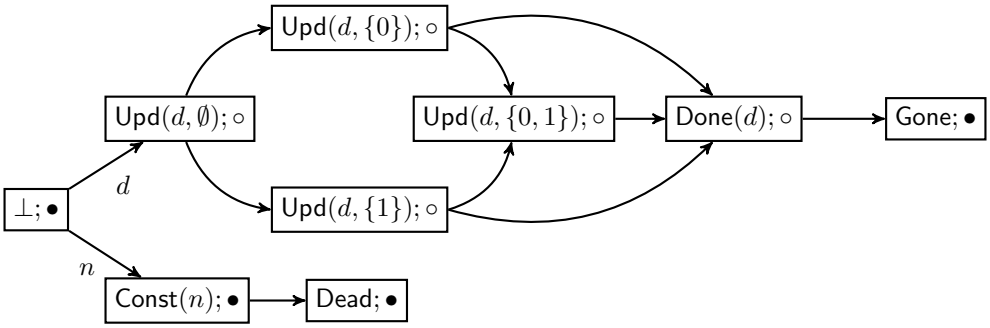
$\langle P \rangle$
  rec $\mathsf{try}()$.
  $\langle P \rangle$
    let $c = \mathsf{head}[1]$ **in**
  $\langle c \propto \mathsf{Sentinel}(-,-) \rangle$
    let $n = \mathsf{getVal}(c)$ **in**
  $\langle n \propto \mathsf{Sentinel}(-,-) \rangle$
    case $n[2]$ **of**
      null $\Rightarrow$ $\langle j \rightarrowtail_{\mathrm{S}} K[\textbf{null}] \rangle$
        null
      | **some**(n') $\Rightarrow$ $\langle P * n \propto \mathsf{Sentinel}(-, n') \rangle$
        if $\mathsf{CAS}(\mathsf{head}[1], c, n')$
        then $\left\langle \begin{array}{l} n \propto \mathsf{Dead}(-, n') * \exists \ell_{\mathrm{I}}, v_{\mathrm{I}}, \ell_{\mathrm{S}}, v_{\mathrm{S}}.\ v_{\mathrm{I}} \preceq^{\mathcal{V}} v_{\mathrm{S}} : \alpha * \\ n' \propto \mathsf{Sentinel}(\ell_{\mathrm{I}}, -) * \ell_{\mathrm{I}} \mapsto_{\mathrm{I}} v_{\mathrm{I}} * \\ j \rightarrowtail_{\mathrm{S}} K[\ell_{\mathrm{S}}] * \ell_{\mathrm{S}} \mapsto_{\mathrm{S}} v_{\mathrm{S}} \end{array} \right\rangle$
          $n'[1]$
        else $\langle P \rangle$
          $\mathsf{try}()$
  $\langle \mathsf{ret}_{\mathrm{I}}.\ \exists \mathsf{ret}_{\mathrm{S}}.\ \mathsf{ret}_{\mathrm{I}} \preceq^{\mathcal{V}} \mathsf{ret}_{\mathrm{S}} : \textbf{ref}_?(\alpha) \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{ret}_{\mathrm{S}}] \rangle$
$\langle \mathsf{ret}_{\mathrm{I}}.\ \exists \mathsf{ret}_{\mathrm{S}}.\ \mathsf{ret}_{\mathrm{I}} \preceq^{\mathcal{V}} \mathsf{ret}_{\mathrm{S}} : \textbf{ref}_?(\alpha) \wedge j \rightarrowtail_{\mathrm{S}} K[\mathsf{ret}_{\mathrm{S}}] \rangle$

5.4 CCAS

$\mathsf{counter_S} \triangleq$
  let $c =$ **new** $0, f =$ **new false**, lock = **new false**
  let get() $=$ sync(lock) { $c[1]$ }
  let setFlag(b) $=$ sync(lock) { $f := b$ }
  let cinc() $=$ **sync**(lock) {
    $c[1] := c[1] +$ **if** $f[1]$ **then** 1 **else** 0 }
  in (get, setFlag, cinc)

$\mathsf{counter_I} \triangleq$
  let $c =$ **new inj**$_1$ $0, f =$ **new false**
  let complete(o, x) $=$
    if $f[1]$ **then** CAS($c, o,$ **inj**$_1$ $(x+1)$)
      else CAS($c, o,$ **inj**$_1$ x)
  let rec get() $=$ **let** $o = c[1]$ **in case** o **of**
    inj$_1$ $x \Rightarrow x$
  | **inj**$_2$ $x \Rightarrow$ complete(o, x); get()
  let setFlag(b) $= f := b$
  let rec cinc() $=$ **let** $o = c[1]$ **in case** o **of**
    inj$_1$ $x \Rightarrow$ **let** $n =$ **inj**$_2$ x **in**
      if CAS(c, o, n) **then** complete(n, x) **else** cinc()
  | **inj**$_2$ $x \Rightarrow$ complete(o, x); cinc()
  in (get, setFlag, cinc)

We have the following "life story" for every "descriptor" (injection into sum) location:



Formally, we have

$$
\begin{array}{rcl}
d & ::= & n, j, K \\
S_0 & \triangleq & \{\bot, \mathsf{Done}(d), \mathsf{Gone}, \mathsf{Const}(n), \mathsf{Dead}\} \cup \{\ \mathsf{Upd}(d, B) \ \mid\ B \subseteq \{0,1\}\ \} \\
S & \triangleq & \left\{\ s \in \mathrm{Loc} \xrightarrow{\mathrm{fin}} S_0 \ \mid\ \exists!\ell.\ (s(\ell) = \mathsf{Const}(-) \vee s(\ell) = \mathsf{Upd}(-))\ \right\} \\
A & \triangleq & \mathrm{Loc}
\end{array}
$$

following the pattern of MSQ. The transition relation on S is the pointwise lifting of that for S_0. Note that the product transition system has one token per location, that is, one token per local transition system. The free tokens F for the product system is the product of the banks for the local systems.

The interpretation is as follows:

$$
\begin{array}{rcl}
d & ::= & n, j, K \qquad\qquad B \subseteq \{0,1\} \qquad\qquad A \triangleq \mathrm{Loc} \\
S_0 & \triangleq & \{\bot, \mathsf{Upd}(d,B), \mathsf{Done}(d), \mathsf{Gone}, \mathsf{Const}(n), \mathsf{Dead}\} \\
S & \triangleq & \left\{ s \in \mathrm{Loc} \xrightarrow{\mathrm{fin}} S_0 \;\middle|\; \exists! \ell.\ s(\ell) \in \{\mathsf{Const}(-), \mathsf{Upd}(-,-)\} \right\} \\
I(s) & \triangleq & \exists b : \mathbf{B}.\ f_{\mathrm{I}} \mapsto_{\mathrm{I}} b * f_{\mathrm{S}} \mapsto_{\mathrm{S}} b * \mathsf{lock} \mapsto_{\mathrm{S}} \mathbf{false} \\
 & & * \begin{cases} \mathsf{linkUpd}(\ell, n, j, K, B) & \exists \ell.\ s(\ell) = \mathsf{Upd}(n, j, K, B) \\ \mathsf{linkConst}(\ell, n) & \exists \ell.\ s(\ell) = \mathsf{Const}(n) \end{cases} \\
 & & * \ast_{s(\ell)=\mathsf{Done}(n,j,K)} \ell \mapsto_{\mathrm{I}} \mathbf{inj}_2\ n * j \rightarrowtail_{\mathrm{S}} K[()] \\
 & & * \ast_{s(\ell)=\mathsf{Gone}} \ell \mapsto_{\mathrm{I}} \mathbf{inj}_2\ - \quad * \quad \ast_{s(\ell)=\mathsf{Dead}} \ell \mapsto_{\mathrm{I}} \mathbf{inj}_1\ - \\
\mathsf{linkConst}(\ell, n) & \triangleq & c_{\mathrm{I}} \mapsto_{\mathrm{I}} \ell * \ell \mapsto_{\mathrm{I}} \mathbf{inj}_1\ n * c_{\mathrm{S}} \mapsto_{\mathrm{S}} n \\
\mathsf{linkUpd}(\ell, n, j, K, B) & \triangleq & c_{\mathrm{I}} \mapsto_{\mathrm{I}} \mathbf{inj}_2\ \ell * \ell \mapsto_{\mathrm{I}} n \\
 & & * \begin{pmatrix} & c_{\mathrm{S}} \mapsto_{\mathrm{S}} n * j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}()] & \\ \oplus & c_{\mathrm{S}} \mapsto_{\mathrm{S}} n * j \rightarrowtail_{\mathrm{S}} K[()] & \text{if } 0 \in B \\ \oplus & c_{\mathrm{S}} \mapsto_{\mathrm{S}} (n+1) * j \rightarrowtail_{\mathrm{S}} K[()] & \text{if } 1 \in B \end{pmatrix}
\end{array}
$$

We let θ be the product transition system.

$$
\begin{array}{lll}
\textbf{let } \mathsf{complete}(o, x) = & & \langle o \propto \mathsf{Upd}(x, -, -, \emptyset) \rangle \\
\quad \textbf{if } f[1] \textbf{ then} & & \langle o \propto \mathsf{Upd}(x, -, -, \{1\}) \rangle \\
\quad\quad \mathsf{CAS}(c, o, \mathbf{inj}_1\ (x+1)) & & \quad \langle o \propto \mathsf{Done}(x, -, -) \rangle \\
\quad \textbf{else} & & \langle o \propto \mathsf{Upd}(x, -, -, \{0\}) \rangle \\
\quad\quad \mathsf{CAS}(c, o, \mathbf{inj}_1\ x) & & \quad \langle o \propto \mathsf{Done}(x, -, -) \rangle \\
 & & \\
\textbf{let rec } \mathsf{cinc}() = & & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * (\theta, I, \emptyset, \emptyset) \rangle \\
 & & \left\langle \begin{array}{l} j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * \\ \left(o \propto \mathsf{Const}(-) \vee o \propto \mathsf{Upd}(-,-) \right) \end{array} \right\rangle \\
\quad \textbf{let } o = c[1] \textbf{ in} & & \\
\quad \textbf{case } o \textbf{ of} & & \\
\quad\quad \mathbf{inj}_1\ x \Rightarrow & & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * o \propto \mathsf{Const}(x) \rangle \\
\quad\quad\quad \textbf{let } n = \mathbf{inj}_2\ x \textbf{ in} & & \left\langle \begin{array}{l} j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * o \propto \mathsf{Const}(x) * \\ n \mapsto \mathbf{inj}_2\ x \end{array} \right\rangle \\
\quad\quad\quad \textbf{if } \mathsf{CAS}(c, o, n) \textbf{ then} & & \langle o \propto \mathsf{Dead}(x) \wedge n \propto_\bullet \mathsf{Upd}(x, j, K, \emptyset) \rangle \\
 & \therefore & \quad \langle n \propto_\bullet \mathsf{Upd}(x, j, K, \emptyset) \rangle \\
\quad\quad\quad\quad \mathsf{complete}(n, x); & & \quad \langle n \propto_\bullet \mathsf{Done}(x, j, K) \rangle \\
\quad\quad\quad\quad () & & \quad \langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \wedge n \propto \mathsf{Gone} \rangle \\
 & \therefore & \langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle \\
\quad\quad\quad \textbf{else} & & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * (\theta, I, \emptyset, \emptyset) \rangle \\
\quad\quad\quad\quad \mathsf{cinc}() & & \quad \langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle \\
 & & \\
\quad\quad | \ \mathbf{inj}_2\ x \Rightarrow & & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * o \propto \mathsf{Upd}(x, -, -, -) \rangle \\
\quad\quad\quad \mathsf{complete}(o, x); & & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * o \propto \mathsf{Done}(x, -, -) \rangle \\
 & \therefore & \langle j \rightarrowtail_{\mathrm{S}} K[\mathsf{cinc}_{\mathrm{S}}()] * (\theta, I, \emptyset, \emptyset) \rangle \\
\quad\quad\quad \mathsf{cinc}() & & \langle \mathsf{ret}.\ \mathsf{ret} = () \wedge j \rightarrowtail_{\mathrm{S}} K[()] \rangle
\end{array}
$$