A EQUIVALENCE OF THE PTSO OPERATIONAL AND DECLARATIVE SEMANTICS

A.1 Intermediate Operational Semantics

Types.

Annotated persistent memory

$$M \in \textsc{AMem} \triangleq \left\{ f \in \textsc{Loc} \xrightarrow{\text{fin}} W \cup U \;\middle|\; \forall \mathtt{x} \in dom(f).\ \mathtt{loc}(f(\mathtt{x})) = \mathtt{x} \right\}$$

Annotated persistent sub-buffers

$$(o, pb) \in \textsc{APSBuff} \triangleq \left\{ (o, pb) \in \textsc{Opt}\,\langle PF \rangle \times \textsc{Loc} \xrightarrow{\text{fin}} \textsc{Seq}\,\langle W \cup U \rangle \;\middle|\; \forall \mathtt{x}, e.\ e \in pb(\mathtt{x}) \Rightarrow \mathtt{loc}(e) = \mathtt{x} \right\}$$

Annotated persistent buffers

$$PB \in \textsc{APBuff} \triangleq \textsc{Seq}\,\langle \textsc{APSBuff} \rangle \setminus \epsilon$$

Annotated volatile buffers

$$b \in \textsc{ABuff} \triangleq \textsc{Seq}\,\langle W \rangle$$

Annotated volatile buffer maps

$$B \in \textsc{ABMap} \triangleq \left\{ B \in \textsc{TId} \xrightarrow{\text{fin}} \textsc{ABuff} \;\middle|\; \forall w.\ \forall \tau \in dom(B).\ w \in B(\tau) \Rightarrow \mathtt{tid}(w) = \tau \right\}$$

Annotated labels

$$\begin{array}{lll}
\textsc{ALabels} \ni \lambda ::= \mathsf{R}\langle r, w \rangle & \text{where } r \in R, w \in W \cup U, \mathtt{loc}(r){=}\mathtt{loc}(w), \mathtt{val_r}(r){=}\mathtt{val_w}(w) \\
\quad \mid \mathsf{U}\langle u, w \rangle & \text{where } u \in U, w \in W \cup U, \mathtt{loc}(u){=}\mathtt{loc}(w), \mathtt{val_r}(u){=}\mathtt{val_w}(w) \\
\quad \mid \mathsf{W}\langle w \rangle & \text{where } w \in W \\
\quad \mid \mathsf{F}\langle f \rangle & \text{where } f \in F \\
\quad \mid \mathsf{PF}\langle pf \rangle & \text{where } pf \in PF \\
\quad \mid \mathsf{PS}\langle ps \rangle & \text{where } ps \in PS \\
\quad \mid \mathsf{B}\langle w \rangle & \text{where } w \in W \\
\quad \mid \mathsf{PB}\langle e \rangle & \text{where } e \in W \cup U \cup PF \\
\quad \mid \mathcal{E}\langle \tau \rangle & \text{where } \tau \in \textsc{TId}
\end{array}$$

$$\pi \in \textsc{Path} \triangleq \textsc{Seq}\left\langle \textsc{ALabels} \setminus \left\{ \mathcal{E}\langle \tau \rangle \;\middle|\; \tau \in \textsc{TId} \right\} \right\rangle \qquad \text{Event paths}$$

$$\pi \in \textsc{PPath} \triangleq \textsc{Seq}\left\langle \textsc{ALabels} \cap \left\{ \mathsf{B}\langle e \rangle, \mathsf{PB}\langle e \rangle \;\middle|\; e \in E \right\} \right\rangle \qquad \text{Propagation paths}$$

$$H \in \textsc{Trace} \triangleq \textsc{PPath} \times \textsc{Path} \qquad \text{Traces}$$

$$\mathcal{H} \in \textsc{Hist} \triangleq \textsc{Seq}\,\langle \textsc{Trace} \rangle \qquad \text{Histories}$$

Let

$$\begin{array}{rl}
\textsc{AMem} \ni M_0 & \text{s.t. } \forall \mathtt{x}.\ M_0(\mathtt{x}) = init_{\mathtt{x}} \text{ with } \mathtt{lab}(init_{\mathtt{x}}) \triangleq \mathtt{W}(\mathtt{x}, 0) \\
\textsc{APSBuff} \ni pb_0 & \text{s.t. } \forall \mathtt{x}.\ \mathrm{pb}_0(\mathtt{x}) = \epsilon \\
\textsc{APBuff} \ni PB_0 & \triangleq (\textsc{None}, pb_0) \\
\textsc{ABuff} \ni b_0 & \triangleq \epsilon \\
\textsc{ABMap} \ni B_0 & \text{s.t. } \forall \tau.\ B_0(\tau) = b_0
\end{array}$$

Storage Subsystem

$$\frac{\mathtt{tid}(w) = \tau}{M, PB, B \xrightarrow{\mathsf{W}\langle w \rangle} M, PB, B[\tau \mapsto w.B(\tau)]} \ (\textsc{AM-Write})$$

$$\frac{B(\tau) = b.w \quad \texttt{loc}(w)=\texttt{x} \quad PB{=}(\textsc{None}, pb).PB'' \quad PB'{=}(\textsc{None}, pb[\texttt{x} \mapsto w.pb(\texttt{x})]).PB''}{M, PB, B \xrightarrow{\mathsf{B}\langle w \rangle} M, PB', B[\tau \mapsto b]} \text{(AM-BProp)}$$

$$\frac{PB{=}PB''.(o, pb) \quad pb(\texttt{x}){=}S.e \quad PB'{=}PB''.(o, pb[\texttt{x} \mapsto S])}{M, PB, B \xrightarrow{\mathsf{PB}\langle e \rangle} M[\texttt{x} \mapsto e], PB', B} \text{(AM-PBProp)}$$

$$\frac{}{M, PB.(\textsc{Some}(pf), pb_0), B \xrightarrow{\mathsf{PB}\langle pf \rangle} M, PB.(\textsc{None}, pb_0), B} \text{(AM-PBPropF)}$$

$$\frac{PB \neq \epsilon}{M, PB.(\textsc{None}, pb_0), B \xrightarrow{\mathcal{E}\langle \tau \rangle} M, PB, B} \text{(AM-PBPropE)}$$

$$\frac{\texttt{tid}(r) = \tau \quad \texttt{loc}(r) = \texttt{x} \quad B(\tau) = b \quad \texttt{read}(M, PB, b, \texttt{x}) = e}{M, PB, B \xrightarrow{\mathsf{R}\langle r, e \rangle} M, PB, B} \text{(AM-Read)}$$

$$\frac{\texttt{tid}(u) = \tau \quad \texttt{loc}(u) = \texttt{x} \quad B(\tau){=}\epsilon \quad PB{=}(\textsc{None}, pb).PB' \quad \texttt{read}(M, PB, \epsilon, \texttt{x}){=}e}{M, PB, B \xrightarrow{\mathsf{U}\langle u, e \rangle} M, (\textsc{None}, pb[\texttt{x} \mapsto u.pb(\texttt{x})]).PB', B} \text{(AM-RMW)}$$

$$\frac{\texttt{tid}(f) = \tau \quad B(\tau) = \epsilon}{M, PB, B \xrightarrow{\mathsf{F}\langle f \rangle} M, PB, B} \text{(AM-Fence)}$$

$$\frac{\texttt{tid}(pf) = \tau \quad B(\tau){=}\epsilon \quad PB = (\textsc{None}, pb).PB'}{M, PB, B \xrightarrow{\mathsf{PF}\langle pf \rangle} M, (\textsc{None}, pb_0).(\textsc{Some}(pf), pb).PB', B} \text{(AM-PFence)}$$

$$\frac{\texttt{tid}(pf) = \tau \quad B(\tau){=}\epsilon}{M, PB_0, B \xrightarrow{\mathsf{PS}\langle ps \rangle} M, PB_0, B} \text{(AM-PSync)}$$

where

$$\texttt{read}(.,.,.,.) : \textsc{AMem} \times \textsc{APBuff} \times \textsc{ABuff} \times \textsc{Loc} \longrightarrow W \cup U$$

$$\texttt{read}(M, PB, b, \texttt{x}) \triangleq \begin{cases} e & \text{if } \texttt{rd}_\texttt{b}(b, \texttt{x}) = e \\ e & \text{if } \texttt{rd}_\texttt{pb}(PB, \texttt{x}) = e \\ M(\texttt{x}) & \text{otherwise} \end{cases}$$

with

$$\texttt{rd}_\texttt{b}(.,.) : \textsc{Seq}\,\langle W \cup U \rangle \times \textsc{Loc} \rightharpoonup E$$

$$\texttt{rd}_\texttt{b}(\epsilon, \texttt{x}) \text{ undef} \qquad \texttt{rd}_\texttt{b}(e.s, \texttt{x}) \triangleq \begin{cases} e & \texttt{loc}(e){=}\texttt{x} \\ \texttt{rd}_\texttt{b}(s, \texttt{x}) & \text{otherwise} \end{cases}$$

$$\texttt{rd}_\texttt{pb}(.,.) : \textsc{APBuff} \times \textsc{Loc} \rightharpoonup W \cup U$$

$$\texttt{rd}_\texttt{pb}(\epsilon, \texttt{x}) \text{ undef} \qquad \texttt{rd}_\texttt{pb}((o, pb).PB, \texttt{x}) \triangleq \begin{cases} e & \text{if } \texttt{rd}_\texttt{b}(pb(\texttt{x}), \texttt{x}) = e \\ \texttt{rd}_\texttt{pb}(PB, \texttt{x}) & \text{otherwise} \end{cases}$$

Thread Subsystem

Thread-local steps.

$$\frac{\mathsf{c}_1, \mathsf{s} \xrightarrow{\lambda} \mathsf{c}_1', \mathsf{s}'}{\mathsf{c}_1; \mathsf{c}_2, \mathsf{s} \xrightarrow{\lambda} \mathsf{c}_1'; \mathsf{c}_2, \mathsf{s}'} \text{(AT-Seq1)} \qquad \frac{}{\textbf{skip}; \mathsf{c}, \mathsf{s} \xrightarrow{\mathcal{E}\langle \tau \rangle} \mathsf{c}, \mathsf{s}} \text{(AT-Seq2)}$$

$$\frac{\mathsf{s}(e) \neq 0}{\textbf{if } e \textbf{ then } \mathsf{c}_1 \textbf{ else } \mathsf{c}_2, \mathsf{s} \xrightarrow{\mathcal{E}\langle \tau \rangle} \mathsf{c}_1, \mathsf{s}} \text{(AT-IfT)}$$

$$\frac{\mathsf{s}(e) = 0}{\textbf{if } e \textbf{ then } \mathsf{c}_1 \textbf{ else } \mathsf{c}_2, \mathsf{s} \xrightarrow{\mathcal{E}\langle\tau\rangle} \mathsf{c}_2, \mathsf{s}} \text{ (AT-IfF)}$$

$$\frac{}{\textbf{while } e \textbf{ do } \mathsf{c}, \mathsf{s} \xrightarrow{\mathcal{E}\langle\tau\rangle} \textbf{if } e \textbf{ then } (\mathsf{c}; \textbf{while } e \textbf{ do } \mathsf{c}) \textbf{ else skip}, \mathsf{s}} \text{ (AT-While)}$$

$$\frac{\mathsf{s}' = \mathsf{s}[a \mapsto \mathsf{s}(e)]}{a := e, \mathsf{s} \xrightarrow{\mathcal{E}\langle\tau\rangle} \textbf{skip}, \mathsf{s}'} \text{ (AT-ReadL)}$$

$$\frac{r = (n, \tau, \mathsf{R}(x, v)) \quad \mathsf{s}' = \mathsf{s}[a \mapsto v]}{a := x, \mathsf{s} \xrightarrow{\mathsf{R}\langle r, w\rangle} \textbf{skip}, \mathsf{s}'} \text{ (AT-Read)} \qquad \frac{w = (n, \tau, \mathsf{W}(x, \mathsf{s}(e)))}{x := e, \mathsf{s} \xrightarrow{\mathsf{W}\langle w\rangle} \textbf{skip}, \mathsf{s}} \text{ (AT-Write)}$$

$$\frac{r = (n, \tau, \mathsf{R}(x, v)) \quad v \neq \mathsf{s}(e) \quad \mathsf{s}' = \mathsf{s}[a \mapsto 0]}{a := \textbf{CAS}(x, e, e'), \mathsf{s} \xrightarrow{\mathsf{R}\langle r, w\rangle} \textbf{skip}, \mathsf{s}'} \text{ (AT-CAS0)}$$

$$\frac{u = (n, \tau, \mathsf{U}(x, \mathsf{s}(e), \mathsf{s}(e'))) \quad \mathsf{s}' = \mathsf{s}[a \mapsto 1]}{a := \textbf{CAS}(x, e, e'), \mathsf{s} \xrightarrow{\mathsf{U}\langle u, w\rangle} \textbf{skip}, \mathsf{s}'} \text{ (AT-CAS1)}$$

$$\frac{u = (n, \tau, \mathsf{U}(x, v, v{+}\mathsf{s}(e))) \quad \mathsf{s}' = \mathsf{s}[a \mapsto v]}{a := \textbf{FAA}(x, e), \mathsf{s} \xrightarrow{\mathsf{U}\langle u, w\rangle} \textbf{skip}, \mathsf{s}'} \text{ (AT-FAA)} \qquad \frac{}{\textbf{fence}, \mathsf{s} \xrightarrow{\mathsf{F}\langle f\rangle} \textbf{skip}, \mathsf{s}} \text{ (AT-Fence)}$$

$$\frac{}{\textbf{pfence}, \mathsf{s} \xrightarrow{\mathsf{PF}\langle pf\rangle} \textbf{skip}, \mathsf{s}} \text{ (AT-PFence)} \qquad \frac{}{\textbf{psync}, \mathsf{s} \xrightarrow{\mathsf{PS}\langle ps\rangle} \textbf{skip}, \mathsf{s}} \text{ (AT-PSync)}$$

Program Steps.

$$\frac{\mathsf{P}(\tau), \mathsf{S}(\tau) \xrightarrow{\lambda} \mathsf{c}, \mathsf{s} \quad \mathtt{tid}(\lambda) = \tau}{\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}[\tau \mapsto \mathsf{c}], \mathsf{S}[\tau \mapsto \mathsf{s}]} \text{ (AP-Step)}$$

where

$$\begin{aligned}
\mathtt{tid}(\mathsf{R}\langle r, w\rangle) &\triangleq \mathtt{tid}(r) \\
\mathtt{tid}(\mathsf{U}\langle u, w\rangle) &\triangleq \mathtt{tid}(u) \\
\mathtt{tid}(\mathsf{W}\langle w\rangle) &\triangleq \mathtt{tid}(w) \\
\mathtt{tid}(\mathsf{F}\langle f\rangle) &\triangleq \mathtt{tid}(f) \\
\mathtt{tid}(\mathsf{PF}\langle pf\rangle) &\triangleq \mathtt{tid}(pf) \\
\mathtt{tid}(\mathsf{PS}\langle ps\rangle) &\triangleq \mathtt{tid}(ps) \\
\mathtt{tid}(\mathsf{B}\langle w\rangle) &\triangleq \mathtt{tid}(w) \\
\mathtt{tid}(\mathsf{PB}\langle e\rangle) &\triangleq \mathtt{tid}(e) \\
\mathtt{tid}(\mathcal{E}\langle\tau\rangle) &\triangleq \tau
\end{aligned}$$

Event-Annotated Operational Semantics

$$\frac{\mathsf{P}, \mathsf{S} \xrightarrow{\mathcal{E}\langle\tau\rangle} \mathsf{P}', \mathsf{S}'}{\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}', \mathsf{S}', M, PB, B, \mathcal{H}, \pi} \ (\text{A-SilentP})$$

$$\frac{M, PB, B \xrightarrow{\mathcal{E}\langle\tau\rangle} M', PB', B'}{\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi} \ (\text{A-SilentM})$$

$$\frac{M, PB, B \xrightarrow{\lambda} M', PB', B' \quad \lambda \in \left\{\mathsf{B}\langle e\rangle, \mathsf{PB}\langle e\rangle\right\} \quad \mathsf{fresh}(\lambda, \pi) \quad \mathsf{fresh}(\lambda, \mathcal{H})}{\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \lambda.\pi} \ (\text{A-PropM})$$

$$\frac{\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}', \mathsf{S}' \quad M, PB, B \xrightarrow{\lambda} M', PB', B' \quad \lambda \neq \mathcal{E}\langle - \rangle \quad \mathsf{fresh}(\lambda, \pi) \quad \mathsf{fresh}(\lambda, \mathcal{H})}{\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}, \lambda.\pi} \ (\text{A-Step})$$

$$\frac{M, PB, B \xrightarrow{\pi'}_{\mathsf{p}} M', PB_0, B_0}{\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathbf{recover}, \mathsf{S}_0, M, PB_0, B_0, ((\pi', \pi).\mathcal{H}), \epsilon} \ (\text{A-Crash})$$

with

$$\frac{}{(M, PB, B) \xrightarrow{\epsilon}_{\mathsf{p}} (M, PB, B)}$$

$$\frac{(M, PB, B) \xrightarrow{\mathsf{PB}\langle e\rangle} (M'', PB'', B'') \quad (M'', PB'', B'') \xrightarrow{\pi}_{\mathsf{p}} (M', PB', B')}{(M, PB, B) \xrightarrow{\mathsf{PB}\langle e\rangle.\pi}_{\mathsf{p}} (M', PB', B')}$$

$$\frac{(M, PB, B) \xrightarrow{\mathsf{B}\langle e\rangle} (M'', PB'', B'') \quad (M'', PB'', B'') \xrightarrow{\pi}_{\mathsf{p}} (M', PB', B')}{(M, PB, B) \xrightarrow{\mathsf{B}\langle e\rangle.\pi}_{\mathsf{p}} (M', PB', B')}$$

and

$$\begin{aligned}\mathsf{fresh}(\lambda, \pi) &\triangleq \lambda \notin \pi \wedge \forall e, w, w'. \\ &\qquad (\lambda{=}\mathsf{R}\langle e, w\rangle \Rightarrow \mathsf{R}\langle e, w'\rangle \notin \pi) \wedge (\lambda{=}\mathsf{U}\langle e, w\rangle \Rightarrow \mathsf{U}\langle e, w'\rangle \notin \pi)\end{aligned}$$

$$\mathsf{fresh}(\lambda, \mathcal{H}) \triangleq \forall (\pi', \pi) \in \mathcal{H}.\ \mathsf{fresh}(\lambda, \pi'.\pi)$$

Definition A.1.

$$\begin{aligned}\mathsf{complete}(\pi) \triangleq \forall e.\ & \mathsf{W}\langle e\rangle \in \pi \Rightarrow \mathsf{B}\langle e\rangle \in \pi \\ & \mathsf{B}\langle e\rangle \in \pi \Rightarrow \mathsf{PB}\langle e\rangle \in \pi \\ & \mathsf{U}\langle e, -\rangle \in \pi \Rightarrow \mathsf{PB}\langle e\rangle \in \pi \\ & \mathsf{PF}\langle e\rangle \in \pi \Rightarrow \mathsf{PB}\langle e\rangle \in \pi\end{aligned}$$

$$
\begin{array}{l}
\mathsf{wfp}(\pi, \mathcal{H}) \triangleq \forall \lambda, \pi_1, \pi_2, e, r, e_1, e_2. \\
\quad \mathsf{nodups}(\pi.\pi'.\pi'') \\
\quad \pi{=}\pi_2.\mathsf{R}\langle r, e\rangle.\pi_1 \lor \pi{=}\pi_2.\mathsf{U}\langle r, e\rangle.\pi_1 \Rightarrow \mathsf{wfrd}(r, e, \pi_1, \pi') \\
\quad \mathsf{B}\langle e\rangle \in \pi \Rightarrow \mathsf{W}\langle e\rangle <_\pi \mathsf{B}\langle e\rangle \\
\quad \mathsf{PB}\langle e\rangle \in \pi \Rightarrow \\
\quad\quad (\mathsf{B}\langle e\rangle <_\pi \mathsf{PB}\langle e\rangle \lor \mathsf{U}\langle e, -\rangle <_\pi \mathsf{PB}\langle e\rangle \lor \mathsf{PF}\langle e\rangle <_\pi \mathsf{PB}\langle e\rangle) \\
\quad \mathtt{tid}(e_1) = \mathtt{tid}(e_2) \Rightarrow \\
\quad\quad \mathsf{B}\langle e_2\rangle \in \pi \land \mathsf{W}\langle e_1\rangle <_\pi \mathsf{W}\langle e_2\rangle \iff \mathsf{B}\langle e_1\rangle <_\pi \mathsf{B}\langle e_2\rangle \\
\quad \mathsf{W}\langle e_1\rangle <_\pi \mathsf{F}\langle e_2\rangle \land \mathtt{tid}(e_1) = \mathtt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1\rangle <_\pi \mathsf{F}\langle e_2\rangle \\
\quad \mathsf{W}\langle e_1\rangle <_\pi \mathsf{U}\langle e_2, e\rangle \land \mathtt{tid}(e_1) = \mathtt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1\rangle <_\pi \mathsf{U}\langle e_2, e\rangle \\
\quad \mathsf{W}\langle e_1\rangle <_\pi \mathsf{PF}\langle e_2\rangle \land \mathtt{tid}(e_1) = \mathtt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1\rangle <_\pi \mathsf{PF}\langle e_2\rangle \\
\quad \mathsf{W}\langle e_1\rangle <_\pi \mathsf{PS}\langle e_2\rangle \land \mathtt{tid}(e_1) = \mathtt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1\rangle <_\pi \mathsf{PS}\langle e_2\rangle \\
\quad \mathtt{loc}(e_1) = \mathtt{loc}(e_2) \land e_1, e_2 \in W \cup U \Rightarrow \\
\quad\quad \mathsf{PB}\langle e_2\rangle \in \pi \land \begin{pmatrix} \mathsf{B}\langle e_1\rangle <_\pi \mathsf{B}\langle e_2\rangle \\ \lor \mathsf{B}\langle e_1\rangle <_\pi \mathsf{U}\langle e_2, -\rangle \\ \lor \mathsf{U}\langle e_1, -\rangle <_\pi \mathsf{B}\langle e_2\rangle \\ \lor \mathsf{U}\langle e_1, -\rangle <_\pi \mathsf{U}\langle e_2, -\rangle \end{pmatrix} \iff \mathsf{PB}\langle e_1\rangle <_\pi \mathsf{PB}\langle e_2\rangle \\
\quad e_1, e_2 \in (PE \times PE) \setminus (W \cup U \times W \cup U) \Rightarrow \\
\quad\quad \mathsf{PB}\langle e_2\rangle \in \pi \land \begin{pmatrix} \mathsf{B}\langle e_1\rangle <_\pi \mathsf{PF}\langle e_2\rangle \\ \lor \mathsf{U}\langle e_1, -\rangle <_\pi \mathsf{PF}\langle e_2\rangle \\ \lor \mathsf{PF}\langle e_1\rangle <_\pi \mathsf{B}\langle e_2\rangle \\ \lor \mathsf{PF}\langle e_1\rangle <_\pi \mathsf{U}\langle e_2, -\rangle \\ \lor \mathsf{PF}\langle e_1\rangle <_\pi \mathsf{PF}\langle e_2\rangle \end{pmatrix} \iff \mathsf{PB}\langle e_1\rangle <_\pi \mathsf{PB}\langle e_2\rangle \\
\quad \begin{pmatrix} \mathsf{B}\langle e_1\rangle <_\pi \mathsf{PS}\langle e_2\rangle \\ \lor \mathsf{U}\langle e_1, -\rangle <_\pi \mathsf{PS}\langle e_2\rangle \\ \lor \mathsf{PF}\langle e_1\rangle <_\pi \mathsf{PS}\langle e_2\rangle \end{pmatrix} \Rightarrow \mathsf{PB}\langle e_1\rangle <_\pi \mathsf{PS}\langle e_2\rangle
\end{array}
$$

where $\pi' = \pi_n. \cdots .\pi_1$ and $\pi'' = \pi'_n. \cdots .\pi'_1$, when $\mathcal{H} = (\pi'_n, \pi_n). \cdots .(\pi'_1, \pi_1)$; and

$$
\mathsf{nodups}(\pi) \triangleq \forall \pi_1, \pi_2, \lambda.\ \pi = \pi_1.\lambda.\pi_2 \Rightarrow \mathsf{fresh}(\lambda, \pi_1.\pi_2)
$$

$$
\mathsf{wfrd}(r, e, \pi, \pi') \triangleq
\begin{pmatrix}
\exists \pi_1, \pi_2, \lambda.\ \pi = \pi_1.\lambda.\pi_2 \\
\quad \land (\lambda{=}\mathsf{B}\langle e\rangle \lor \lambda{=}\mathsf{U}\langle e, -\rangle \lor (\lambda{=}\mathsf{W}\langle e\rangle \land \mathtt{tid}(e) = \mathtt{tid}(r))) \\
\quad \land \begin{pmatrix} (\lambda{=}\mathsf{B}\langle e\rangle \lor \lambda{=}\mathsf{U}\langle e, -\rangle) \Rightarrow \\ \left\{\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi_1 \;\middle|\; \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset \\ \land \left\{ e' \;\middle|\; \begin{array}{l} \mathsf{W}\langle e'\rangle \in \pi \land \mathsf{B}\langle e'\rangle \notin \pi \\ \land\, \mathtt{loc}(e'){=}\mathtt{loc}(r) \land \mathtt{tid}(e'){=}\mathtt{tid}(r) \end{array} \right\} = \emptyset \end{pmatrix} \\
\quad \land \begin{pmatrix} \lambda{=}\mathsf{W}\langle e\rangle \Rightarrow \\ \mathsf{B}\langle e\rangle \notin \pi_1 \land \left\{ \mathsf{W}\langle e'\rangle \in \pi_1 \;\middle|\; \begin{array}{l} \mathtt{loc}(e'){=}\mathtt{loc}(r) \land \\ \mathtt{tid}(e'){=}\mathtt{tid}(r) \end{array} \right\} = \emptyset \end{pmatrix}
\end{pmatrix}
$$

$$
\lor \begin{pmatrix} \exists \pi_1, \pi_2.\ \pi' = \pi_1.\mathsf{PB}\langle e\rangle.\pi_2 \\ \quad \land \left\{ \begin{array}{l} \mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi, \\ \mathsf{W}\langle e''\rangle \in \pi, \\ \mathsf{PB}\langle e'\rangle \in \pi_1 \end{array} \;\middle|\; \begin{array}{l} \mathtt{loc}(e'){=}\mathtt{loc}(r) \land \\ \mathtt{loc}(e''){=}\mathtt{loc}(r) \land \\ \mathtt{tid}(e''){=}\mathtt{tid}(r) \end{array} \right\} = \emptyset \end{pmatrix}
$$

$$
\lor \begin{pmatrix} e = init_{\mathtt{loc}(e)} \land \left\{ \begin{array}{l} \mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi, \\ \mathsf{W}\langle e''\rangle \in \pi, \\ \mathsf{PB}\langle e'\rangle \in \pi' \end{array} \;\middle|\; \begin{array}{l} \mathtt{loc}(e'){=}\mathtt{loc}(r) \land \\ \mathtt{loc}(e''){=}\mathtt{loc}(r) \land \\ \mathtt{tid}(e''){=}\mathtt{tid}(r) \end{array} \right\} = \emptyset \end{pmatrix}
$$

Definition A.2.

$$\begin{aligned}\mathtt{wf}(M, PB, B, \mathcal{H}, \pi) \overset{\text{def}}{\iff} & \ \mathtt{mem}(\mathcal{H}, \pi) = M \land \mathtt{pbuff}(PB_0, \pi) \land \mathtt{bmap}(B_0, \pi) \\ & \land \mathsf{wfp}(\pi, \mathcal{H}) \land \mathsf{wfh}(\mathcal{H})\end{aligned}$$

where

$$\mathtt{mem}(\mathcal{H}, \pi) = M \overset{\text{def}}{\iff} \forall \mathtt{x} \in \textsc{Loc}.\ M(\mathtt{x}) = \mathtt{read}(\mathcal{H}, \pi, \mathtt{x})$$

$$\begin{aligned}\mathtt{read}(\mathcal{H}, \lambda.\pi, \mathtt{x}) &\triangleq \begin{cases} e & \exists e.\ \lambda = \mathsf{PB}\langle e\rangle \land \mathtt{loc}(e) = \mathtt{x} \\ \mathtt{read}(\mathcal{H}, \pi, \mathtt{x}) & \text{otherwise}\end{cases} \\ \mathtt{read}((-, \pi).\mathcal{H}, \epsilon, \mathtt{x}) &\triangleq \mathtt{read}(\mathcal{H}, \pi, \mathtt{x}) \\ \mathtt{read}(\epsilon, \epsilon, \mathtt{x}) &\triangleq init_{\mathtt{x}}\end{aligned}$$

$$\mathtt{pbuff}(PB, \epsilon) \triangleq PB$$

$$\mathtt{pbuff}((\textsc{None}, pb).PB, \pi.\lambda) \triangleq \begin{cases} \mathtt{pbuff}((\textsc{None}, pb[\mathtt{x} \mapsto e.pb(\mathtt{x})]).PB, \pi) & \text{if } \exists e, \mathtt{x}. \\ & \quad \lambda \in \{\mathsf{U}\langle e, -\rangle, \mathsf{B}\langle e\rangle\} \\ & \quad \land \mathtt{loc}(e){=}\mathtt{x} \\ & \quad \land \mathsf{PB}\langle e\rangle \notin \pi \\ \mathtt{pbuff}((\textsc{None}, pb_0).(\textsc{Some}(e), pb).PB, \pi) & \text{if } \exists e.\ \lambda = \mathsf{PF}\langle e\rangle \\ & \quad \land \mathsf{PB}\langle e\rangle \notin \pi \\ \mathtt{pbuff}((\textsc{None}, pb).PB, \pi) & \text{otherwise}\end{cases}$$

$$\begin{aligned}\mathtt{bmap}(B, \epsilon) &\triangleq B \\ \mathtt{bmap}(B, \pi.\lambda) &\triangleq \begin{cases} \mathtt{bmap}(B[\tau \mapsto e.B(\tau)], \pi) & \text{if } \exists e, \mathtt{x}.\ \lambda = \mathsf{W}\langle e\rangle \land \mathtt{tid}(e){=}\tau \\ & \quad \land \mathsf{B}\langle e\rangle \notin \pi \\ \mathtt{bmap}(B, \pi) & \text{otherwise}\end{cases}\end{aligned}$$

$$\begin{aligned}\mathsf{wfh}(\epsilon) &\overset{\text{def}}{\iff} true \\ \mathsf{wfh}((\pi', \pi).\mathcal{H}) &\overset{\text{def}}{\iff} \mathsf{wfp}(\pi'.\pi, \mathcal{H}) \land \mathsf{complete}(\pi'.\pi) \land \mathsf{wfh}(\mathcal{H})\end{aligned}$$

Lemma A.1. *For all* $\mathsf{P}, \mathsf{P}', \mathsf{S}, \mathsf{S}', PB, PB', B, B', \mathcal{H}, \mathcal{H}', \pi, \pi'$:

- $\mathsf{wf}(M_0, PB_0, B_0, \epsilon, \epsilon)$
- *if* $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}', \pi'$ *and* $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$, *then* $\mathsf{wf}(M', PB', B', \mathcal{H}', \pi')$
- *if* $\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \mathbf{skip}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi$, *then* $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$

Proof. The proof of the first part follows trivially from the definitions of M_0, PB_0, and B_0. The second part follows straightforwardly by induction on the structure of $\Rightarrow$. The last part follows from the previous two parts and induction on the length of $\Rightarrow^*$. □

Graph Operational Semantics

Let

$$\Gamma \in \textsc{GHist} \triangleq \textsc{Seq}\,\langle \textsc{Graph} \times \textsc{Trace}\rangle \ \text{Graph histories}$$

$$\frac{\mathsf{P}, \mathsf{S} \xrightarrow{\mathcal{E}\langle\tau\rangle} \mathsf{P}', \mathsf{S}'}{\mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \mathsf{P}', \mathsf{S}', \Gamma, \pi}\ (\textsc{G-SilentP})$$

$$\frac{\lambda \in \{\mathsf{B}\langle e\rangle, \mathsf{PB}\langle e\rangle\} \quad \mathsf{fresh}(\lambda, \pi) \quad \mathsf{fresh}(\lambda, \Gamma)}{\mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \mathsf{P}, \mathsf{S}, \Gamma, \lambda.\pi}\ (\textsc{G-Prop})$$

$$\frac{\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}', \mathsf{S}' \quad \lambda \neq \mathcal{E}\langle - \rangle \quad \mathsf{fresh}(\lambda, \pi) \quad \mathsf{fresh}(\lambda, \Gamma)}{\mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \mathsf{P}', \mathsf{S}', \Gamma, \lambda.\pi} \text{ (G-Step)}$$

$$\frac{\texttt{comp}(\pi, \pi') \quad \texttt{getG}(\Gamma, \pi, \pi') = G}{\mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \textbf{recover}, \mathsf{S}_0, (G, (\pi', \pi)).\Gamma, \epsilon} \text{ (G-Crash)}$$

where

$$\begin{array}{rcl}
\mathsf{fresh}(\lambda, \Gamma) & \stackrel{\text{def}}{\iff} & \forall (-, (\pi', \pi)) \in \Gamma.\ \mathsf{fresh}(\lambda, \pi'.\pi) \\
\texttt{comp}(.,.) & : & \textsc{Path} \times \textsc{PPath} \longrightarrow \{true, false\} \\
\texttt{comp}(\pi, \pi') & \stackrel{\text{def}}{\iff} & \forall e.\ \mathsf{W}\langle e \rangle \in \pi \wedge \mathsf{B}\langle e \rangle \notin \pi \iff \mathsf{B}\langle e \rangle \in \pi' \\
& & \wedge \begin{pmatrix} (\mathsf{W}\langle e \rangle \in \pi \wedge \mathsf{PB}\langle e \rangle \notin \pi) \\ \vee (\mathsf{U}\langle e, - \rangle \in \pi \wedge \mathsf{PB}\langle e \rangle \notin \pi) \\ \vee (\mathsf{PF}\langle e \rangle \in \pi \wedge \mathsf{PB}\langle e \rangle \notin \pi) \end{pmatrix} \iff \mathsf{PB}\langle e \rangle \in \pi'
\end{array}$$

$$\texttt{getG}(\Gamma, \pi, \pi') \triangleq \begin{cases} (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo}) & \text{if } \mathsf{wfp}(\pi'.\pi, \texttt{hist}(\Gamma)) \wedge \mathsf{complete}(\pi'.\pi) \\ \text{undefined} & \text{otherwise} \end{cases}$$

with

$$\texttt{hist}(\epsilon) = \epsilon \qquad \texttt{hist}((G, H).\Gamma) = H.\texttt{hist}(\Gamma)$$

$$\begin{array}{l}
E^0 = \begin{cases} \left\{ init_{\mathsf{x}} \,\middle|\, \mathsf{x} \in \mathsf{Loc} \right\} & \text{if } \Gamma = \epsilon \\ \left\{ \max\left(G.\mathsf{nvo}|_{G.E^P \cap (U_{\mathsf{x}} \cup W_{x})} \right) \,\middle|\, \mathsf{x} \in \mathsf{Loc} \right\} & \text{if } \Gamma = (G, -).\Gamma' \end{cases} \\
E^P = E^0 \cup \left\{ e \,\middle|\, \exists \lambda \in \pi.\ \texttt{getPE}(\lambda) = e \right\} \\
E = E^0 \cup \left\{ e \,\middle|\, \exists \lambda \in \pi.\ \texttt{getE}(\lambda) = e \right\} \\
\mathsf{rf} = \left\{ (w, e) \,\middle|\, \mathsf{R}\langle e, w \rangle \in \pi \vee \mathsf{U}\langle e, w \rangle \in \pi \right\} \\
\mathsf{po} = E^0 \times (E \setminus E^0) \cup \bigcup_{\tau \in \textsc{TId}} \left\{ (e_1, e_2) \,\middle|\, \begin{array}{l} \exists \lambda_1, \lambda_2 \in \pi. \\ \quad e_1 = \texttt{getE}(\lambda_1) \wedge e_2 = \texttt{getE}(\lambda_2) \\ \quad \wedge\, \texttt{tid}(e_1) = \texttt{tid}(e_2) = \tau \\ \quad \wedge\, \lambda_1 <_{\pi} \lambda_2 \end{array} \right\} \\
\mathsf{tso} \triangleq E^0 \times (E \setminus E^0) \\
\qquad \cup \left\{ (e_1, e_2) \,\middle|\, \begin{array}{l} \exists \lambda_1, \lambda_2 \in \pi'.\pi. \\ \quad e_1 = \texttt{getBE}(\lambda_1) \wedge e_2 = \texttt{getBE}(\lambda_2) \wedge \lambda_1 <_{\pi'.\pi} \lambda_2 \end{array} \right\} \\
\mathsf{nvo} \triangleq E^0 \times (E \setminus E^0) \\
\qquad \cup \left\{ (e_1, e_2) \,\middle|\, \begin{array}{l} \exists \lambda_1, \lambda_2 \in \pi'.\pi. \\ \quad e_1 = \texttt{getPE}(\lambda_1) \wedge e_2 = \texttt{getPE}(\lambda_2) \wedge \lambda_1 <_{\pi'.\pi} \lambda_2 \end{array} \right\}
\end{array}$$

and

$$\texttt{getE}(.) : \textsc{ALabels} \rightharpoonup E$$
$$\texttt{getE}(\lambda) \triangleq \begin{cases} e & \text{if } \exists e, w.\ \lambda \in \{\mathsf{R}\langle e, w\rangle, \mathsf{U}\langle e, w\rangle, \mathsf{W}\langle e\rangle, \mathsf{F}\langle e\rangle, \mathsf{PF}\langle e\rangle, \mathsf{PS}\langle e\rangle\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\texttt{getPE}(.) : \textsc{ALabels} \rightharpoonup E$$
$$\texttt{getPE}(\lambda) \triangleq \begin{cases} e & \text{if } \exists e.\ \lambda \in \{\mathsf{R}\langle e,,\rangle \mathsf{F}\langle e\rangle, \mathsf{PS}\langle e\rangle, \mathsf{PB}\langle e\rangle\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\texttt{getBE}(.) : \textsc{ALabels} \rightharpoonup E$$
$$\texttt{getBE}(\lambda) \triangleq \begin{cases} e & \text{if } \exists e, w.\ \lambda \in \{\mathsf{R}\langle e, w\rangle, \mathsf{U}\langle e, w\rangle, \mathsf{F}\langle e\rangle, \mathsf{PF}\langle e\rangle, \mathsf{PS}\langle e\rangle, \mathsf{B}\langle e\rangle\} \\ \text{undefined} & \text{otherwise} \end{cases}$$

A.2 Soundness of the Intermediate Semantics against PTSO Declarative Semantics

Theorem 4 (soundness). *For all* $\mathsf{P}, \mathsf{S}, M, \mathcal{H} = (\pi_{n-1}, \pi'_{n-1}). \cdots .(\pi_1, \pi'_1), \pi_n$ *and* $\pi'_n = \epsilon$*:*

$$\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \mathbf{skip} || \cdots || \mathbf{skip}, \mathsf{S}, M, PB_0, B_0, \mathcal{H}, \pi_n$$

then

(1) $\mathsf{P}, \mathsf{S}_0, \epsilon, \epsilon \Rightarrow^* \mathbf{skip} || \cdots || \mathbf{skip}, \mathsf{S}, \Gamma, \pi_n$ *where*

$$\Gamma = \Gamma_n$$
$$\Gamma_1 = \epsilon \qquad \Gamma_{j+1} = (G_j, (\pi'_j, \pi_j)). \cdots .(G_1, (\pi'_1, \pi_1)) \quad \textit{for } j \in \{1 \cdots n{-}1\}$$
$$G_i = \texttt{getG}(\Gamma_i, \pi_i, \pi'_i) \qquad \textit{for } i \in \{1 \cdots n\}$$

(2) $\mathcal{E} = G_1; \cdots; G_n$ *is PTSO-valid.*

Proof. Pick arbitrary $\mathsf{P}, \mathsf{S}, M, \mathcal{H} = (\pi_{n-1}, \pi'_{n-1}). \cdots .(\pi_1, \pi'_1), \pi_n$ such that

$$\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \mathbf{skip} || \cdots || \mathbf{skip}, \mathsf{S}, M, PB_0, B_0, \mathcal{H}, \pi_n$$

and let $\pi'_n = \epsilon$. The proof of the first part follows from Lemma A.1 and by induction on the length of the event-annotated transition $\Rightarrow^*$.

For the second part, for all $i \in \{1 \cdots n\}$ let $E_i = R_i \cup F_i \cup PE_i$ with $PE_i = W_i \cup U_i \cup PF_i \cup PS_i$. As $G_i = \texttt{getG}(\Gamma_i, \pi_i, \pi'_i)$, we know that $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ hold. It then suffices to show that for all $i \in \{1 \cdots n\}$ and $G_i = (E^0_i, E^P_i, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$:

$$E^0_i \subseteq E^P_i \tag{1}$$
$$E^P_i \subseteq E_i \tag{2}$$
$$E^0_i \times (E_i \setminus E^0_i) \subseteq \mathsf{po}_i \tag{3}$$
$$E^0_i \times (E_i \setminus E^0_i) \subseteq \mathsf{tso}_i \tag{4}$$
$$E^0_i \times (E_i \setminus E^0_i) \subseteq \mathsf{nvo}_i \tag{5}$$
$$dom(\mathsf{nvo}_i; [E^P_i]) \subseteq E^P_i \text{ and } E^P_n = E_n \tag{6}$$
$$E^0_0 = \left\{ init_{\mathtt{x}} \,\middle|\, \mathtt{x} \in \mathrm{Loc} \right\} \text{ and } E^0_{i+1} = \left\{ \max\left(\mathsf{nvo}_i|_{E^P_i \cap (U_{\mathtt{x}} \cup W_x)} \right) \,\middle|\, \mathtt{x} \in \mathrm{Loc} \right\} \tag{7}$$
$$R_i \cup F_i \cup PS_i \subseteq E^P_i \text{ and } \mathsf{po}_i; [PS_i] \subseteq E^P_i \tag{8}$$
$$\mathsf{po}_i \text{ is a strict total order on } E_i \tag{9}$$

$$\mathsf{rf}_i \subseteq (W_i \cup U_i) \times (R_i \cup U_i) \text{ and is total and functional on } R_i \cup U_i \tag{10}$$

$$\mathsf{tso}_i \subseteq E_i \times E_i \text{ and is total on } E_i \setminus R_i \tag{11}$$

$$\mathsf{po}_i \setminus (W_i \times R_i) \subseteq \mathsf{tso}_i \tag{12}$$

$$\mathsf{rf}_i \subseteq \mathsf{tso}_i \cup \mathsf{po}_i \tag{13}$$

$$\forall (w, r) \in \mathsf{rf}_i.\ \forall w' \in W_i \cup U_i.\ (w', r) \in \mathsf{tso}_i \cup \mathsf{po}_i \land \mathtt{loc}(w'){=}\mathtt{loc}(r) \Rightarrow (w, w') \notin \mathsf{tso}_i \tag{14}$$

$$\mathsf{nvo}_i \text{ is a strict total order on } PE_i \tag{15}$$

$$\forall \mathrm{x} \in \mathrm{Loc}.\ (\mathsf{nvo}_i)_{\mathrm{x}} \subseteq \mathsf{tso}_i \tag{16}$$

$$[PS_i]; \mathsf{tso}_i; [PE_i] \ \cup\ [PE_i]; \mathsf{tso}_i; [PS_i] \subseteq \mathsf{nvo}_i \tag{17}$$

$$[PF_i]; \mathsf{tso}_i; [PE_i] \ \cup\ [PE_i]; \mathsf{tso}_i; [PF_i] \subseteq \mathsf{nvo}_i \tag{18}$$

The proofs of parts (1), (3), (4), (5), (7), and (9) follow immediately from the construction of G_i.

RTS. (2)
Pick an arbitrary $e \in E^P_i$. We then know there exist $\lambda \in \pi_i$ and e such that $e = \mathtt{getPE}(\lambda)$ and either $\lambda = \mathsf{R}\langle e, -\rangle$, or $\lambda = \mathsf{F}\langle e\rangle$, or $\lambda = \mathsf{PS}\langle e\rangle$, or $\lambda = \mathsf{PB}\langle e\rangle$. In the first three cases, from the definition of $\mathtt{getE}(.)$ we know that $e = \mathtt{getE}(\lambda)$ and thus from the definition of E_i we have $e \in E_i$, as required. In the last case, from $\mathsf{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ we know that there exists w such that either $\mathsf{W}\langle e\rangle \in \pi_i$, or $\mathsf{U}\langle e, w\rangle \in \pi_i$, or $\mathsf{PF}\langle e\rangle \in \pi_i$. As such, from the definition of E_i we have $e \in E_i$, as required.

RTS. (6)
Pick an arbitrary e_1, e_2 such that $(e_1, e_2) \in \mathsf{nvo}_i$ and $e_2 \in E^P_i$. From the definition of nvo_i we then know there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \mathtt{getPE}(\lambda_1)$, $e_2 = \mathtt{getPE}(\lambda_2)$ and $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$. On the other hand, from the definition of E^P_i and since $e_2 \in E^P_i$ we know that $\lambda_2 \in \pi_i$. As such, since $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$ and labels in $\pi'_i.\pi_i$ are fresh ($\mathsf{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ holds), we also know that $\lambda_1 \in \pi_i$. Consequently, since $e_1 = \mathtt{getPE}(\lambda_1)$ and $\lambda_1 \in \pi_i$, from the definition of E^P_i we have $e_1 \in E^P_i$, as required.

To demonstrate that $E^P_n = E_n$, it suffices to show that $E_n \subseteq E^P_n$, as in part (2) we established that $E^P_n \subseteq E_n$. Pick arbitrary $e \in E_n$. From the definition of E_n we then know there exists $\lambda \in \pi_n$ such that $\mathtt{getE}(\lambda) = e$. There are then two cases to consider: 1) $e \notin W_n \cup U_n \cup PF_n$; or 2) $e \in W_n \cup U_n \cup PF_n$. In case (1) from the definition of $\mathtt{getPE}(.)$ we know that $\mathtt{getPE}(\lambda) = e$ and thus $e \in E^P_n$, as required. In case (2) from $\mathsf{complete}(\pi'_n.\pi_n)$ we know that there exists λ' such that $\lambda' = \mathsf{PB}\langle e\rangle$ and $\lambda' \in \pi'_n.\pi_n$. As $\pi'_n = \epsilon$ we know that $\lambda' \in \pi_n$. As such, from the definition of $\mathtt{getPE}(.)$ we know that $\mathtt{getPE}(\lambda') = e$ and thus $e \in E^P_n$, as required.

RTS. (8)
The proof of the first part follows immediately from the definitions of E^P_i and $\mathtt{getPE}(.)$. For the second part, pick an arbitrary $(e, ps) \in \mathsf{po}_i; [PS_i]$, i.e. $(e, ps) \in \mathsf{po}_i$ and $ps \in PS_i$. From the definition of po_i we then know there exist $\lambda, \lambda' \in \pi_i$ such that $e = \mathtt{getE}(\lambda)$, $\lambda' = \mathsf{PS}\langle ps\rangle$, $ps = \mathtt{getE}(\lambda')$, $\lambda \prec_{\pi_i} \lambda'$, and $\mathtt{tid}(e) = \mathtt{tid}(ps)$. There are now two cases to consider: 1) $e \notin U_i \cup W_i \cup PF_i$; or 2) $e \in U_i \cup W_i \cup PF_i$.

In case (1) from the definition of $\mathtt{getPE}(.)$ we have $\mathtt{getPE}(\lambda) = e$ and thus from the definition of E^P_i we have $e \in E^P_i$, as required.

In case (2) from $\mathsf{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ we know there exists $\lambda'' = \mathsf{PB}\langle e\rangle$ such that $\lambda \prec_{\pi_i} \lambda'' \prec_{\pi_i} \lambda'$. That is, $\lambda'' \in \pi_i$. As such, such from the definition of E^P_i we have $e \in E^P_i$, as required.

RTS. (10)
To demonstrate that $\mathsf{rf}_i \subseteq (W_i \cup U_i) \times (R_i \cup U_i)$, pick an arbitrary $(e_w, e_r) \in \mathsf{rf}_i$. From the definition of rf_i we then know there exists $\lambda \in \pi_i$ such that $\lambda = \mathsf{R}\langle e_r, e_w \rangle$ or $\lambda = \mathsf{U}\langle e_r, e_w \rangle$. As such from the type of annotated labels we know $e_r \in R \cup U$ and $e_w \in W \cup U$.

To demonstrate that rf_i is total on R_i, pick an arbitrary $r \in R_i$. Form the definition of E_i we then know there exist $\lambda \in \pi_i$ and e such that $\lambda = \mathsf{R}\langle r, e \rangle$. As such we know $(e, r) \in \mathsf{rf}_i$ and thus rf_i is *total* on R_i. The proof of rf_i being total on U_i is analogous and omitted here.

To show rf_i is functional on R_i, pick an arbitrary $r \in R_i$. Form the definition of E_i we know there exists $\lambda \in \pi_i$ and e such that $\lambda = \mathsf{R}\langle r, e \rangle$. As such we know $(e, r) \in \mathsf{rf}_i$ and thus rf_i. Moreover, since π_i contains unique labels ($\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ holds), we know $\forall e' {\neq} e.\ \mathsf{R}\langle r, e' \rangle \notin \pi_i$ and thus $\forall e' {\neq} e.\ (e', r) \notin \mathsf{rf}_i$. That is, rf_i is *functional* on R_i. The proof of rf_i being functional on U_i is analogous and omitted here.

RTS. (11)
To demonstrate that $\mathsf{tso}_i \subseteq E_i \times E_i$, pick an arbitrary $(e_1, e_2) \in \mathsf{tso}_i$. From the definition of tso_i we then know there exists $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \mathtt{getBE}(\lambda_1)$ and $e_2 = \mathtt{getBE}(\lambda_2)$. For $j \in \{1, 2\}$, we then know that either 1) $\neg \exists e.\ \lambda_j = \mathsf{B}\langle e \rangle$; or 2) $\lambda_j = \mathsf{B}\langle e_j \rangle$. In case (1) since $\pi'_i \in \textsc{PPath}$ we know that $\lambda_j \in \pi_i$ and thus from the definition of E_i we know $e_j \in E_i$. In case (2) from $\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ we know that $\mathsf{W}\langle e_j \rangle \in \pi'_i.\pi_i$. As such, since $\pi'_i \in \textsc{PPath}$, we know that $\mathsf{W}\langle e_j \rangle \in \pi_i$ and thus from the definition of E_i we have $e_j \in E_i$. As such, in both cases we have $(e_1, e_2) \in E_i \times E_i$, as required.

Transitivity and strictness of tso_i follow from the definition of tso_i, transitivity and strictness of $<_{\pi'_i.\pi_i}$ and the freshness of events in $\pi'_i.\pi_i$ ($\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$holds).

To demonstrate that tso_i is total on $E_i \setminus R_i$, pick arbitrary $e_1, e_2 \in E_i \setminus R_i$ such that $e_1 \neq e_2$. For $j \in \{1, 2\}$, from the definitions of E_i we know there exist $\lambda_j \in \pi_i$ such that either 1) $e_j \in E_i \setminus (R_i \cup W_i)$ and $\lambda_j = \mathtt{getE}(\lambda_j)$; or 2) $e_j \in W_i$ and $\lambda_j = \mathsf{W}\langle e_j \rangle$. In case (1) we then have $\lambda_j \in \pi'_i.\pi_i$ and $\mathtt{getBE}(\lambda_j) = e_j$. In case (2) from $\text{complete}(\pi'_i.\pi_i)$ we then know there exists $\lambda'_j = \mathsf{B}\langle e_j \rangle \in \pi'_i.\pi_i$ and $\mathtt{getBE}(\lambda'_j) = e_j$. As such, in both cases we know there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \mathtt{getBE}(\lambda_1)$ and $e_2 = \mathtt{getBE}(\lambda_2)$. As $e_1 \neq e_2$ and $\pi'_j.\pi_j$ contains fresh labels ($\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ holds), we know that $\lambda_1 \neq \lambda_2$ and thus either $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$ or $\lambda_2 <_{\pi'_i.\pi_i} \lambda_1$. As such, from the definition of tso_i we have either $(e_1, e_2) \in \mathsf{tso}_i$ or $(e_2, e_1) \in \mathsf{tso}_i$, as required.

RTS. (12)
Pick an arbitrary $(e_1, e_2) \in \mathsf{po}_i \setminus (W_i \times R_i)$. From the definition of po_i we then know there exist τ and $\lambda_1, \lambda_2 \in \pi_i$ such that $e_1 = \mathtt{getE}(\lambda_1)$, $e_2 = \mathtt{getE}(\lambda_2)$, $\mathtt{tid}(e_1) = \mathtt{tid}(e_2) = \tau$ and $\lambda_1 <_{\pi_i} \lambda_2$. That is, $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$. There are then three cases to consider: $e_1, e_2 \notin W_i$; or 2) $e_1 \notin W_i \wedge e_2 \in W_i$; or 3) $e_1 \in W_i$.

In case (1) from the definition of $\mathtt{getBE}(.)$ we know that $e_1 = \mathtt{getBE}(\lambda_1)$, $e_2 = \mathtt{getBE}(\lambda_2)$. As such, from the definition of tso_i we have $(e_1, e_2) \in \mathsf{tso}_i$.

In case (2), from the definition of $\mathtt{getBE}(.)$ we know that $e_1 = \mathtt{getBE}(\lambda_1)$. On the other hand, from $\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ and $\text{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{B}\langle e_2 \rangle$ such that $\lambda_2 <_{\pi'_i.\pi_i} \lambda$. That is, $e_2 = \mathtt{getBE}(\lambda)$. Since we also have $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$, from the transitivity of $<$ we have $\lambda_1 <_{\pi'_i.\pi_i} \lambda$. As such, from the definition of tso_i we have $(e_1, e_2) \in \mathsf{tso}_i$, as required.

In case (3) , there are three additional cases to consider: i) $\lambda_2 = \mathsf{F}\langle e_2 \rangle$ or $\lambda_2 = \mathsf{PF}\langle e_2 \rangle$ or $\lambda_2 = \mathsf{U}\langle e_2, - \rangle$; or ii) $\lambda_2 = \mathsf{W}\langle e_2 \rangle$; or iii) $\lambda_2 = \mathsf{PS}\langle e_2 \rangle$.

In case (3.i) from the definition of $\mathtt{getBE}(.)$ we know that $e_2 = \mathtt{getBE}(\lambda_2)$. On the other hand, from $\text{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ and $\text{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{B}\langle e_1 \rangle$ such that $\lambda_1 <_{\pi'_i.\pi_i}$

$\lambda \prec_{\pi'_i.\pi_i} \lambda_2$. That is, $e_1 = \texttt{getBE}(\lambda)$. As such, from the definition of tso_i we have $(e_1, e_2) \in \text{tso}_i$, as required.

In case (3.ii) from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we know there exist $\lambda'_1 = \mathsf{B}\langle e_1 \rangle$ and $\lambda'_2 = \mathsf{B}\langle e_2 \rangle$ such that $\lambda'_1 \prec_{\pi'_i.\pi_i} \lambda'_2$. That is, $e_1 = \texttt{getBE}(\lambda'_1)$ and $e_2 = \texttt{getBE}(\lambda'_2)$. As such, from the definition of tso_i we have $(e_1, e_2) \in \text{tso}_i$, as required.

In case (3.iii) from the definition of $\texttt{getBE}(.)$ we know that $e_2 = \texttt{getBE}(\lambda_2)$. On the other hand, from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ and since $\texttt{tid}(e_1) = \texttt{tid}(e_2)$, we know there exist $\lambda'_1 = \mathsf{B}\langle e_1 \rangle$ such that $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda'_1 \prec_{\pi'_i.\pi_i} \mathsf{PB}\langle e_1 \rangle \prec_{\pi'_i.\pi_i} \lambda_2$. That is, $e_1 = \texttt{getBE}(\lambda'_1)$. As such, from the definition of tso_i we have $(e_1, e_2) \in \text{tso}_i$, as required.

RTS. (13)
Pick arbitrary $(w, r) \in \text{rf}_i$. From the construction of rf_i we then know there exist $\lambda \in \pi_i$ such that either $\lambda = \mathsf{R}\langle r, w \rangle$ or $\lambda = \mathsf{U}\langle r, w \rangle$. From $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ we then know that either 1) $\mathsf{B}\langle w \rangle \prec_{\pi_i} r$; or 2) $\mathsf{U}\langle w, - \rangle \prec_{\pi_i} r$; or 3) $\mathsf{W}\langle w \rangle \prec_{\pi_i} r$ and $\texttt{tid}(w) = \texttt{tid}(r)$; or 4) $w \in E^0_i$. In cases (1-2) from the definition of tso_i we have $(w, r) \in \text{tso}_i$, as required. In cases (3-4) from the definition of po_i we have $(w, r) \in \text{po}_i$, as required.

RTS. (14)
Pick arbitrary $(w, r) \in \text{rf}_i$ and $w' \in U_i \cup W_i$ such that $(w', r) \in \text{tso}_i \cup \text{po}_i$ and $\texttt{loc}(w') = \texttt{loc}(r)$. If $w' = w$, from the strictness of tso_i we immediately know that $(w, w') \notin \text{tso}_i$, as required.

Now let us consider the case where $w' \neq w$. From the construction of rf_i we then know there exist $\lambda \in \pi_i$ such that either $\lambda_r = \mathsf{R}\langle r, w \rangle$ or $\lambda_r = \mathsf{U}\langle r, w \rangle$. From $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ we then know that either 1) there exists $\lambda = \mathsf{B}\langle w \rangle \prec_{\pi_i} \lambda_r$; or 2) there exists $\lambda = \mathsf{U}\langle w, - \rangle \prec_{\pi_i} \lambda_r$; or 3) there exists $\lambda = \mathsf{W}\langle w \rangle \prec_{\pi_i} \lambda_r$ and $\texttt{tid}(w) = \texttt{tid}(r)$; or 4) $w \in E^0_i$.

On the other hand, from the construction of tso_i, po_i and since $(w', r) \in \text{tso}_i \cup \text{po}_i$ we know that either: a) there exists $\lambda' = \mathsf{B}\langle w' \rangle \prec_{\pi_i} r$; or b) there exists $\lambda' = \mathsf{U}\langle w', - \rangle \prec_{\pi_i} r$; or c) $w' \in E^0_i$.

However, from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and since $\lambda = \mathsf{R}\langle r, w \rangle \in \pi_i$ or $\lambda = \mathsf{U}\langle r, w \rangle \in \pi_i$, in cases (1.a), (1.b), (2.1), (2.b), (3.a), (3.b) we have $\lambda' \prec_{\pi_i} \lambda$. Consequently, in cases (1.a), (1.b), (2.1), (2.b) from the definition of tso_i we have $(w', w) \in \text{tso}_i$, i.e. $(w, w') \notin \text{tso}_i$, as required. In cases (3.a) and (3.b) from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we additionally know there exist $\lambda'' = \mathsf{B}\langle w \rangle$ such that $\lambda \prec_{\pi'_i.\pi_i} \lambda''$ and thus from the transitivity of $\prec$ we have $\lambda' \prec_{\pi'_i.\pi_i} \lambda''$. Consequently, from the definition of tso_i we have $(w', w) \in \text{tso}_i$, i.e. $(w, w') \notin \text{tso}_i$, as required.

In cases (2.c), (3.c) from the definition of tso_i we have $(w', w) \in \text{tso}_i$, i.e. $(w, w') \notin \text{tso}_i$, as required. Similarly, in case (1.c) from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ we know $\mathsf{W}\langle w \rangle \in \pi_i$ and thus from the definition of tso_i we have $(w', w) \in \text{tso}_i$, i.e. $(w, w') \notin \text{tso}_i$, as required.

Cases (4.1), (4.b) cannot arise as from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ we arrive at a contradiction. Case (4.c) cannot arise as $w \neq w'$ and from the definition of E^0_i we cannot have two distinct events of the same location in E^0_i.

RTS. (15)
Transitivity and strictness of nvo_i follow from the definition of nvo_i, transitivity and strictness of $\prec_{\pi'_i.\pi_i}$ and the freshness of events in $\pi'_i.\pi_i$ ($\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$holds).

To demonstrate that nvo_i is total on PE_i, pick arbitrary $e_1, e_2 \in PE_i$ such that $e_1 \neq e_2$. For $j \in \{1, 2\}$, from the definitions of PE_i we know there exist $\lambda_j \in \pi_i$ such that either 1)$e_j \in U_i$ and $\lambda_j = \mathsf{U}\langle e_j, - \rangle$; or 2) $e_j \in W_i$ and $\lambda_j = \mathsf{W}\langle e_j \rangle$; or 3) $e_j \in PF_i$ and $\lambda_j = \mathsf{PF}\langle e_j \rangle$; or 4) $e_j \in PS_i$ and $\lambda_j = \mathsf{PS}\langle e_j \rangle$. In cases (1-3) from $\texttt{complete}(\pi'_i.\pi_i)$ we then know there exists $\lambda'_j = \mathsf{PB}\langle e_j \rangle \in \pi'_i.\pi_i$ and $\texttt{getPE}(\lambda'_j) = e_j$. In case (4) we have $\texttt{getPE}(\lambda_j) = e_j$. As such, in both cases we know there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getPE}(\lambda_1)$ and $e_2 = \texttt{getPE}(\lambda_2)$. As $e_1 \neq e_2$ and $\pi'_j.\pi_j$ contains fresh labels ($\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ holds), we know that $\lambda_1 \neq \lambda_2$ and thus either $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$ or $\lambda_2 <_{\pi'_i.\pi_i} \lambda_1$. As such, from the definition of nvo_i we have either $(e_1, e_2) \in \mathsf{nvo}_i$ or $(e_2, e_1) \in \mathsf{nvo}_i$, as required.

RTS. (16)
Pick arbitrary $\texttt{x} \in \text{Loc}$ and $(e_1, e_2) \in (\mathsf{nvo}_i)_\texttt{x}$. From the definition of nvo_i we then know there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getPE}(\lambda_1)$, $e_2 = \texttt{getPE}(\lambda_2)$, $\texttt{loc}(e_1) = \texttt{loc}(e_2) = \texttt{x}$, $\lambda_1 <_{\pi_i} \lambda_2$, $e_1, e_2 \in W_i \cup U_i$ and $\lambda_1 = \mathsf{PB}\langle e_1 \rangle$ and $\lambda_2 = \mathsf{PB}\langle e_2 \rangle$. From $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ we then know that either 1) $e_1, e_2 \in W_i$ and $\mathsf{B}\langle e_1 \rangle <_{\pi'_i.\pi_i} \mathsf{B}\langle e_2 \rangle$; or 2) $e_1, e_2 \in U_i$ and there exist e'_1, e'_2 such that $\mathsf{U}\langle e_1, e'_1 \rangle <_{\pi'_i.\pi_i} \mathsf{U}\langle e_2, e'_2 \rangle$; or 3) $e_1 \in W$, $e_2 \in U_i$ and there exists e'_2 such that $\mathsf{B}\langle e_1 \rangle <_{\pi'_i.\pi_i} \mathsf{U}\langle e_2, e'_2 \rangle$; or 4) $e_1 \in U_i$, $e_2 \in W_i$ and there exists e'_1 such that $\mathsf{U}\langle e_1, e'_1 \rangle <_{\pi'_i.\pi_i} \mathsf{B}\langle e_2 \rangle$. In all four cases from the definition of tso_i we have $(e_1, e_2) \in \mathsf{tso}_i$, as required.

RTS. (17)
To demonstrate $[PS_i]; \mathsf{tso}_i; [PE_i] \subseteq \mathsf{nvo}_i$, pick arbitrary $(e_1, e_2) \in [PS_i]; \mathsf{tso}_i; [PE_i]$. From the definition of tso_i we then know that that there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getBE}(\lambda_1)$, $e_2 = \texttt{getBE}(\lambda_2)$ and $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$. Moreover, since $e_1 \in PS_i$ we know that $\texttt{getPE}(\lambda_1) = e_1$. There are now three cases to consider: 1) $e_2 \notin W_i \cup U_i \cup PF_i$; or 2) $e_2 \in U_i \cup PF_i$; or 3) $e_2 \in W_i$.

In case (1), from the definitions of $\texttt{getPE}(.)$ and $\texttt{getBE}(.)$ we know that $\texttt{getPE}(\lambda_2) = e_2$ and thus from the definition of nvo_i we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

In case (2) from the definition of $\texttt{getBE}(.)$ we know that either $\lambda_2 = \mathsf{U}\langle e_2, - \rangle$ or $\lambda_2 = \mathsf{PF}\langle e_2 \rangle$ and thus from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{PB}\langle e_2 \rangle$ such that $\lambda_2 <_{\pi'_i.\pi_i} \lambda$. Since we also have $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$, from the transitivity of $<_{\pi'_i.\pi_i}$ we also have $\lambda_1 <_{\pi'_i.\pi_i} \lambda$. Moreover, from the definition of $\texttt{getPE}(.)$ we have $\texttt{getPE}(\lambda) = e_2$. Consequently, we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

Similarly, in case (3) from the definition of $\texttt{getBE}(.)$ we know $\lambda_2 = \mathsf{B}\langle e_2 \rangle$ and thus from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{PB}\langle e_2 \rangle$ such that $\lambda_2 <_{\pi'_i.\pi_i} \lambda$. Since we also have $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$, from the transitivity of $<_{\pi'_i.\pi_i}$ we also have $\lambda_1 <_{\pi'_i.\pi_i} \lambda$. Moreover, from the definition of $\texttt{getPE}(.)$ we have $\texttt{getPE}(\lambda) = e_2$. Consequently, we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

To demonstrate $[PE_i]; \mathsf{tso}_i; [PS_i] \subseteq \mathsf{nvo}_i$, pick arbitrary $(e_1, e_2) \in [PE_i]; \mathsf{tso}_i; [PS_i]$. From the definition of tso_i we then know that that there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getBE}(\lambda_1)$, $e_2 = \texttt{getBE}(\lambda_2)$ and $\lambda_1 <_{\pi'_i.\pi_i} \lambda_2$. Moreover, since $e_2 \in PS_i$ we know that $\texttt{getPE}(\lambda_2) = e_2$. There are now four cases to consider: 1) $e_1 \notin W_i \cup U_i \cup PF_i$; or 2) $e_1 \in U_i$; or 3) $e_1 \in W_i$; or 4) $e_1 \in PF_i$.

In case (1), from the definitions of $\texttt{getPE}(.)$ and $\texttt{getBE}(.)$ we know that $\texttt{getPE}(\lambda_1) = e_1$ and thus from the definition of nvo_i we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

In case (2) from the definition of $\texttt{getBE}(.)$ we know $\lambda_1 = \mathsf{U}\langle e_1, - \rangle$ and thus from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{PB}\langle e_1 \rangle$ such that $\lambda_1 <_{\pi'_i.\pi_i} \lambda <_{\pi'_i.\pi_i} \lambda_2$. Moreover, from the definition of $\texttt{getPE}(.)$ we have $\texttt{getPE}(\lambda) = e_1$. Consequently, we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

Similarly, in case (3) from the definition of $\texttt{getBE}(.)$ we know $\lambda_1 = \mathsf{B}\langle e_1 \rangle$ and thus from $\texttt{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\texttt{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{PB}\langle e_1 \rangle$ such that $\lambda_1 <_{\pi'_i.\pi_i} \lambda <_{\pi'_i.\pi_i} \lambda_2$. Moreover, from the definition of $\texttt{getPE}(.)$ we have $\texttt{getPE}(\lambda) = e_1$. Consequently, we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

Analogously, in case (4) from the definition of $\texttt{getBE}(.)$ we know $\lambda_1 = \mathsf{PF}\langle e_1\rangle$ and thus from $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda = \mathsf{PB}\langle e_1\rangle$ such that $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda \prec_{\pi'_i.\pi_i} \lambda_2$. Moreover, from the definition of $\texttt{getPE}(.)$ we have $\texttt{getPE}(\lambda) = e_1$. Consequently, we have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

RTS. (18)
To demonstrate that $[PE_i]; \mathsf{tso}_i; [PF_i] \subseteq \mathsf{nvo}_i$, pick an arbitrary $(e_1, e_2) \in [PE_i]; \mathsf{tso}_i; [PF_i]$. If $e_1 \in PS_i$, then the desired result holds immediately from part (17). On the other hand if $e_1 \notin PS_i$, then from the definition of tso_i we then know that that there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getBE}(\lambda_1)$, $e_2 = \texttt{getBE}(\lambda_2)$, $\lambda_2 = \mathsf{PF}\langle e_2\rangle$, $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$ and either 1) $\lambda_1 = \mathsf{B}\langle e_1\rangle$; 2) $\lambda_1 = \mathsf{U}\langle e_1, -\rangle$; or 3) $\lambda_1 = \mathsf{PF}\langle e_1\rangle$. From $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda'_2 = \mathsf{PB}\langle e_2\rangle$ such that $\lambda_2 \prec_{\pi'_i.\pi_i} \lambda'_2$. As such we have $\texttt{getPE}(\lambda'_2) = e_2$. As $\lambda_2 = \mathsf{PF}\langle e_2\rangle$ and $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$, in all three cases from $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ we know there exist $\lambda'_1 = \mathsf{PB}\langle e_1\rangle$ such that $\lambda'_1 \prec_{\pi'_i.\pi_i} \lambda'_2$. That is, $\texttt{getPE}(\lambda'_1) = e_1$. From the definition of nvo_i we thus have $(e_1, e_2) \in \mathsf{nvo}_i$, as required.

Similarly, to demonstrate that $[PF_i]; \mathsf{tso}_i; [PE_i] \subseteq \mathsf{nvo}_i$, pick an arbitrary $(e_1, e_2) \in [PF_i]; \mathsf{tso}_i; [PE_i]$. If $e_2 \in PS_i$, then the desired result holds immediately from part (17). On the other hand if $e_2 \notin PS_i$, then from the definition of tso_i we then know that that there exist $\lambda_1, \lambda_2 \in \pi'_i.\pi_i$ such that $e_1 = \texttt{getBE}(\lambda_1)$, $e_2 = \texttt{getBE}(\lambda_2)$, $\lambda_1 = \mathsf{PF}\langle e_1\rangle$, $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$ and either 1) $\lambda_2 = \mathsf{B}\langle e_2\rangle$; 2) $\lambda_2 = \mathsf{U}\langle e_2, -\rangle$; or 3) $\lambda_2 = \mathsf{PF}\langle e_2\rangle$. From $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ we know there exists $\lambda'_1 = \mathsf{PB}\langle e_1\rangle \in \pi'_i.\pi_i$. As such we have $\texttt{getPE}(\lambda'_1) = e_1$. As $\lambda_1 = \mathsf{PF}\langle e_1\rangle$ and $\lambda_1 \prec_{\pi'_i.\pi_i} \lambda_2$, in all three cases from $\mathsf{wfp}(\pi'_i.\pi_i, \texttt{hist}(\Gamma_i))$ and $\mathsf{complete}(\pi'_i.\pi_i)$ we know there exist $\lambda'_2 = \mathsf{PB}\langle e_2\rangle$ such that $\lambda'_1 \prec_{\pi'_i.\pi_i} \lambda'_2$. That is, $\texttt{getPE}(\lambda'_2) = e_2$. From the definition of nvo_i we thus have $(e_1, e_2) \in \mathsf{nvo}_i$, as required. □

A.3 Completeness of the Intermediate Semantics against PTSO Declarative Semantics

Definition A.3. Let $\mathcal{E} = G_1; \cdots; G_n$ denote a PTSO-valid execution chain. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. For each execution era G_i, the set of traces induced by G_i, written $\texttt{traces}(G_i, S_i)$, includes those traces (π', π) that satisfy the following condition:

$$(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \texttt{traces}(G_i, S_i) \overset{\text{def}}{\iff} \bigwedge_{k=1}^{i} \texttt{getG}(\Gamma_k, \pi_k, \pi'_k) = G_k$$

where $\Gamma_1 = \epsilon$ and $\Gamma_{j+1} = (\pi'_j, \pi_j). \cdots .(\pi'_1, \pi_1)$ for $j \in \{1 \cdots i{-}1\}$.

Lemma A.2. *Let $\mathcal{E} = G_1; \cdots; G_n$ denote a PTSO-valid execution chain. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. For all $i \in \{1 \cdots n\}$, $\texttt{traces}(G_i, S_i) \neq \emptyset$.*

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. For an arbitrary PTSO-valid G_i, we demonstrate how to construct a trace $s = (\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1)$ such that $s \in \texttt{traces}(G_i, S_i)$.

For each $k \in \{1 \cdots i\}$ and $G_k = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$,we construct (π'_k, π_k) as follows. Let $R = \{r_1 \cdots r_q\}$ denote an enumeration of $G_k.R$ and $\{w_1, \cdots, w_s\}$ denote an enumeration of $G_k.W$. For each $j \in \{1 \cdots q\}$ and $l \in \{0 \cdots s{-}1\}$ where $(w, r_j) \in \mathsf{rf}$, we then define

$$\mathsf{tso}_j^{l+1} \triangleq \begin{cases} \left(\mathsf{tso}_j^l \cup \left\{(r_j, w_{l+1})\right\}\right)^+ & \text{if } (r_j, w_{l+1}) \notin \mathsf{tso}_j^l \cup (\mathsf{tso}_j^l)^{-1} \\ & \text{and } (w, w_{l+1}) \in \mathsf{tso} \\ \mathsf{tso}_j^l & \text{otherwise} \end{cases}$$

where $\text{tso}_1^0 = \text{tso}$ and $\text{tso}_{j+1}^0 = \text{tso}_j^s$ for $j \in \{1 \cdots q{-}1\}$. Note that each tso_j^l is 1) total on writes and respects with tso; and 2) is a strict order on E. We next show that:

$$\begin{array}{l}\forall j \in \{1 \cdots q\}.\ \forall l \in \{0 \cdots s\}.\ \forall w, r.\ \forall w' \in W \cup U. \\ \quad (w, r) \in \text{rf} \land (w', r) \in \text{tso}_j^l \cup \text{po} \land \texttt{loc}(w) = \texttt{loc}(w') \Rightarrow (w, w') \notin \text{tso}_j^l\end{array} \tag{RFJ}$$

Let $(w, r_j) \in \text{rf}$. We proceed by double induction on j and l.

Base case $j = 1$ and $l = 0$
As G_k is PTSO-valid, we know that the desired property holds of tso and thus of $\text{tso}_1^0 = \text{tso}$ by definition.

Inductive case $j = 1$ and $l = a{+}1$ with $0 \leq a < s$

$$\begin{array}{l}\forall l' \in \{1 \cdots a\}.\ \forall w, r.\ \forall w' \in W \cup U. \\ \quad (w, r) \in \text{rf} \land (w', r) \in \text{tso}_1^{l'} \cup \text{po} \land \texttt{loc}(w) = \texttt{loc}(w') \Rightarrow (w, w') \notin \text{tso}_1^{l'}\end{array} \tag{I.H.}$$

From the definition of tso_1^l, we know that either i) $\text{tso}_1^l = \text{tso}_1^a$; or ii) $\text{tso}_1^l = \left(\text{tso}_1^a \cup \{(r_1, w_l)\}\right)^+$, $(r_1, w_l) \notin \text{tso}_1^a \cup (\text{tso}_1^a)^{-1}$ and $(w, w_l) \in \text{tso}$. In case (i) the desired result holds immediately from (I.H.).

In case (ii) we proceed by contradiction. Let us assume there exists w_c, w'_c, r_c such that $(w_c, r_c) \in \text{rf}$, $(w'_c, r_c) \in \text{tso}_1^l \cup \text{po} \land \texttt{loc}(w_c) = \texttt{loc}(w'_c)$ and $(w_c, w'_c) \in \text{tso}_1^l$. As $(w_c, w'_c) \in \text{tso}_1^l$ and tso_1^l is a strict order, we know that $w_c \neq w'_c$. On the other hand, from (I.H.) we then know that $(w'_c, r_c) \notin \text{tso}_1^a \cup \text{po}$. As such, form the definition of tso_1^l we know that $w'_c \xrightarrow{\text{tso}_1^a} r_1 \xrightarrow{\text{tso}_1^l} w_l \xrightarrow{\text{tso}_1^a} r_c$. However, as tso_1^a is strict and is total on writes, we know that either a) $(w_l, w'_c) \in \text{tso}_1^a$; or b) $(w'_c, w_l) \in \text{tso}_1^a$. In case (ii.a) we then have $w_l \xrightarrow{\text{tso}_1^a} w'_c \xrightarrow{\text{tso}_1^a} r_1$, contradicting the assumption that $(r_1, w_l) \notin \text{tso}_1^a \cup (\text{tso}_1^a)^{-1}$. In case (ii.b) we have $w'_c \xrightarrow{\text{tso}_1^a} w_l \xrightarrow{\text{tso}_1^a} r_c$, i.e. $(w'_c, r_c) \in \text{tso}_1^a$. As such, from (I.H.) we have $(w_c, w'_c) \notin \text{tso}_1^a$, i.e. $(w'_c, w_c) \in \text{tso}_1^a \subseteq \text{tso}_1^l$, and thus $(w_c, w'_c) \notin \text{tso}_1^l$, contradicting our assumption that $(w_c, w'_c) \in \text{tso}_1^l$.

Inductive case $j = b{+}1$ and $l = 0$ with $1 \leq b < q{-}1$

$$\begin{array}{l}\forall j' \in \{1 \cdots b\}.\ \forall l' \in \{1 \cdots s\}.\ \forall w, r.\ \forall w' \in W \cup U. \\ \quad (w, r) \in \text{rf} \land (w', r) \in \text{tso}_{j'}^{l'} \Rightarrow (w, w') \notin \text{tso}_{j'}^{l'}\end{array} \tag{I.H.}$$

As $\text{tso}_j^0 \triangleq \text{tso}_b^s$, the desired result holds immediately from (I.H.).

Inductive case $j = b{+}1$ and $l = a{+}1$ with $1 \leq b < q{-}1$ and $0 \leq a < s$

$$\begin{array}{l}\forall l' \in \{1 \cdots a\}.\ \forall w, r.\ \forall w' \in W \cup U. \\ \quad (w, r) \in \text{rf} \land (w', r) \in \text{tso}_j^{l'} \Rightarrow (w, w') \notin \text{tso}_j^{l'}\end{array} \tag{I.H.}$$

From the definition of tso_j^l, we know that either i) $\text{tso}_j^l = \text{tso}_j^a$; or ii) $\text{tso}_j^l = \left(\text{tso}_j^a \cup \{(r_j, w_l)\}\right)^+$, $(r_j, w_l) \notin \text{tso}_j^a \cup (\text{tso}_j^a)^{-1}$ and $(w, w_l) \in \text{tso}$. In case (i) the desired result holds immediately from (I.H.).

In case (ii), we proceed by contradiction. Let us assume there exists w_c, w'_c, r_c such that $(w_c, r_c) \in \text{rf}$, $(w'_c, r_c) \in \text{tso}_j^l \cup \text{po} \land \texttt{loc}(w_c) = \texttt{loc}(w'_c)$ and $(w_c, w'_c) \in \text{tso}_j^l$. As $(w_c, w'_c) \in \text{tso}_j^l$ and tso_j^l is a strict order, we know that $w_c \neq w'_c$. On the other hand, from (I.H.) we then know that $(w'_c, r_c) \notin \text{tso}_j^a \cup \text{po}$. As such, form the definition of tso_j^l we know that $w'_c \xrightarrow{\text{tso}_j^a} r_j \xrightarrow{\text{tso}_j^l} w_l \xrightarrow{\text{tso}_j^a} r_c$.

However, as tso^a_j is strict and is total on writes, we know that either a) $(w_l, w'_c) \in \mathsf{tso}^a_j$; or b) $(w'_c, w_l) \in \mathsf{tso}^a_j$. In case (ii.a) we then have $w_l \xrightarrow{\mathsf{tso}^a_j} w'_c \xrightarrow{\mathsf{tso}^a_j} r_j$, contradicting the assumption that $(r_j, w_l) \notin \mathsf{tso}^a_j \cup (\mathsf{tso}^a_j)^{-1}$. In case (ii.b) we have $w'_c \xrightarrow{\mathsf{tso}^a_j} w_l \xrightarrow{\mathsf{tso}^a_j} r_c$, i.e. $(w'_c, r_c) \in \mathsf{tso}^a_j$. As such, from (I.H.) we have $(w_c, w'_c) \notin \mathsf{tso}^a_j$, i.e. $(w'_c, w_c) \in \mathsf{tso}^a_j \subseteq \mathsf{tso}^l_j$, and thus $(w_c, w'_c) \notin \mathsf{tso}^l_j$, contradicting our assumption that $(w_c, w'_c) \in \mathsf{tso}^l_j$. □

Let tso_t denote an extension of tso^s_q to a strict total order on E. Once again, we demonstrate that:

$$\forall w, r.\ \forall w' \in W \cup U.\ (w, r) \in \mathsf{rf} \wedge (w', r) \in \mathsf{tso}_t \wedge \mathtt{loc}(w) = \mathtt{loc}(w') \Rightarrow (w, w') \notin \mathsf{tso}_t \qquad \text{(RF)}$$

Pick arbitrary w, w', r such that $(w, r) \in \mathsf{rf} \wedge \mathtt{loc}(w) = \mathtt{loc}(w')$ and $(w', r) \in \mathsf{tso}_t$. There are two cases to consider: 1) $(w', r) \in \mathsf{tso}^s_q$; or 2) $(w', r) \in \mathsf{tso}_t \setminus \mathsf{tso}^s_q$. In case (1) the result holds from (RFJ) established above. In case (2), as tso_t is a strict order we know that $(r, w') \notin \mathsf{tso}_t$ and thus $(r, w') \notin \mathsf{tso}^s_q$. Moreover, as $(w', r) \in \mathsf{tso}_t \setminus \mathsf{tso}^s_q$, i.e. $(w', r) \notin \mathsf{tso}^s_q$. As such, from the definition of tso^s_q we know that $(w, w') \notin \mathsf{tso}$, i.e. $(w', w) \in \mathsf{tso} \subseteq \mathsf{tso}_t$. As tso_t is a strict order, we have $(w, w') \notin \mathsf{tso}_t$. □

Let $\{e_1, \cdots, e_n\}$ denote an enumeration of $G_k.E \setminus E^0$ that respects tso_t; $\{w_1, \cdots, w_m\}$ denote an enumeration of $G_k.W \setminus E^0$ that respects tso; and $\{e'_1, \cdots, e'_o\}$ denote an enumeration of $G_k.(W \cup U \cup PF) \setminus E^0$ that respects nvo. Since G_k is PTSO-valid and thus $dom(\mathsf{nvo}; [E^P]) \subseteq E^P$, we know there exists p such that $0 \leq p \leq o$ and $\{e'_1, \cdots, e'_p\} \in E^P \setminus E^0$ and $\{e'_{p+1}, \cdots, e'_o\} \in E \setminus (E^P \cup E^0)$.

Let $\pi^0 = \lambda_n. \cdots .\lambda_1$, where $\lambda_j = \mathtt{genBL}(e_j, G_k)$ for $j \in \{1 \cdots n\}$ and:

$$\mathtt{genBL}(e, G) \triangleq \begin{cases} \mathsf{B}\langle e \rangle & \text{if } e \in G.W \\ \mathtt{genL}(e, G) & \text{if } e \in G.E \setminus G.W \\ \text{undefined} & \text{otherwise} \end{cases}$$

For each $j \in \{1 \cdots m\}$, let $N_j = \left\{e \,\middle|\, (w_j, e) \in \mathsf{po} \wedge e \notin \{w_{j+1} \cdots w_m\}\right\}$; and $n_j = \min(\mathsf{po}|_{N_j})$ when such an element exists. For each $j \in \{1 \cdots m\}$, let $\pi^j = \mathtt{addW}(\pi^{j-1}, w_j, n_j)$, where:

$$\mathtt{addW}(\pi, w, n) \triangleq \begin{cases} \mathsf{W}\langle w \rangle.\mathsf{B}\langle w \rangle.s & \text{if } \exists s.\ \pi = \mathsf{B}\langle w \rangle.s \\ \mathsf{W}\langle w \rangle.n.s & \text{if } \exists s.\ \pi = \mathtt{genL}(n, G_k).s \\ e.\mathtt{addW}(s, w, n) & \text{if } \exists s.\ \pi = e.s \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathtt{genL}(e, G) \triangleq \begin{cases} \mathsf{R}\langle e, e' \rangle & \text{if } e \in G.R \wedge (e', e) \in G.\mathsf{rf} \\ \mathsf{W}\langle e \rangle & \text{if } e \in G.W \\ \mathsf{U}\langle e, e' \rangle & \text{if } e \in G.U \wedge (e', e) \in G.\mathsf{rf} \\ \mathsf{F}\langle e \rangle & \text{if } e \in G.F \\ \mathsf{PF}\langle e \rangle & \text{if } e \in G.PF \\ \mathsf{PS}\langle e \rangle & \text{if } e \in G.PS \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that for all $j \in \{1 \cdots m\}$, the $\mathtt{addW}(\pi^{j-1}, w_j, n_j)$ is always defined as $\mathsf{B}\langle w_j \rangle \in \pi^{j-1}$.

For each $j \in \{1 \cdots p\}$, let $P_j = \left\{e \,\middle|\, (e, e'_j) \in \textsf{nvo}\right\}$; and $p_j = \max\left(\textsf{nvo}|_{P_j}\right)$ when such an element exists. Let $\hat{\pi}^0 = \pi^m$ and for each $j \in \{1 \cdots p\}$, let $\hat{\pi}^j = \texttt{addP}(\hat{\pi}^{j-1}, e'_j, p_j)$, where:

$$\texttt{addP}(\pi, e, p) \triangleq \begin{cases} s.\texttt{genPL}(e, G_i).\texttt{genBL}(e, G_i) & \text{if } \exists s.\ \pi = s.\texttt{genBL}(e, G_i) \\ s.\texttt{genPL}(e, G_i).\texttt{genPL}(p, G_i) & \text{if } \exists s.\ \pi = s.\texttt{genPL}(p, G_i) \\ \texttt{addP}(s, e, p).e' & \text{if } \exists s, e'.\ \pi = s.e' \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\texttt{genPL}(e, G) \triangleq \begin{cases} \mathsf{PB}\langle e \rangle & \text{if } e \in G.(W \cup U \cup PF) \\ \texttt{genL}(e, G) & \text{if } e \in G.PS \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that for all $j \in \{1 \cdots p\}$, the $\texttt{addP}(\hat{\pi}^{j-1}, e'_j, p_j)$ is always defined as $\texttt{genBL}(e'_j, G_k) \in \hat{\pi}^{j-1}$. Let $\pi_k = \hat{\pi}^p$ and let $\pi'_k = \texttt{genPL}(e_o, G_k). \cdots .\texttt{genPL}(e_{p+1}, G_k)$.

We next demonstrate that $\texttt{wfp}(\pi'_k.\pi_k, \texttt{hist}(\Gamma_k))$ and $\texttt{complete}(\pi'_k.\pi_k)$ hold.

Goal: $\texttt{wfp}(\pi'_k.\pi_k, \texttt{hist}(\Gamma_k))$
Let $\pi = \pi'_k.\pi_k$. We are then required to show that for all $\lambda, \pi_1, \pi_2, e, r, e_1, e_2$:

$$\texttt{nodups}(\pi.\pi''.\pi''') \tag{19}$$

$$\pi{=}\pi_2.\mathsf{R}\langle r, e \rangle.\pi_1 \lor \pi{=}\pi_2.\mathsf{U}\langle r, e \rangle.\pi_1 \Rightarrow \mathsf{wfrd}(r, e, \pi_1, \pi'') \tag{20}$$

$$\mathsf{B}\langle e \rangle \in \pi \Rightarrow \mathsf{W}\langle e \rangle <_\pi \mathsf{B}\langle e \rangle \tag{21}$$

$$\mathsf{PB}\langle e \rangle \in \pi \Rightarrow (\mathsf{B}\langle e \rangle <_\pi \mathsf{PB}\langle e \rangle \lor \mathsf{U}\langle e, - \rangle <_\pi \mathsf{PB}\langle e \rangle \lor \mathsf{PF}\langle e \rangle <_\pi \mathsf{PB}\langle e \rangle) \tag{22}$$

$$\texttt{tid}(e_1) = \texttt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_2 \rangle \in \pi \land \mathsf{W}\langle e_1 \rangle <_\pi \mathsf{W}\langle e_2 \rangle \iff \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle \tag{23}$$

$$\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{F}\langle e_2 \rangle \land \texttt{tid}(e_1) = \texttt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{F}\langle e_2 \rangle \tag{24}$$

$$\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, e \rangle \land \texttt{tid}(e_1) = \texttt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, e \rangle \tag{25}$$

$$\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle \land \texttt{tid}(e_1) = \texttt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle \tag{26}$$

$$\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \land \texttt{tid}(e_1) = \texttt{tid}(e_2) \Rightarrow \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \tag{27}$$

$$\texttt{loc}(e_1) = \texttt{loc}(e_2) \land e_1, e_2 \in W \cup U \Rightarrow \mathsf{PB}\langle e_2 \rangle \in \pi \land \begin{pmatrix} \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle \\ \lor \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle \\ \lor \mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{B}\langle e_2 \rangle \\ \lor \mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{U}\langle e_2, - \rangle \end{pmatrix} \iff \mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle \tag{28}$$

$$e_1, e_2 \in (PE \times PE) \setminus (W \cup U \times W \cup U) \Rightarrow \mathsf{PB}\langle e_2 \rangle \in \pi \land \begin{pmatrix} \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle \\ \lor \mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{PF}\langle e_2 \rangle \\ \lor \mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle \\ \lor \mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle \\ \lor \mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle \end{pmatrix} \iff \mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle \tag{29}$$

$$\begin{pmatrix} \mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \\ \lor \mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \\ \lor \mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \end{pmatrix} \Rightarrow \mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle \tag{30}$$

where $\pi'' = \pi_{k-1}.\cdots.\pi_1$ and $\pi''' = \pi'_{k-1}.\cdots.\pi'_1$.

The proof of parts (19), (21), (22) follow immediately from the constructions of π'_k and π_k.

For part (20), pick arbitrary π_1, π_2, r, e such that $\pi{=}\pi_2.\mathsf{R}\langle r, e\rangle.\pi_1$ or $\pi{=}\pi_2.\mathsf{U}\langle r, e\rangle.\pi_1$. From the construction of π we then know that $(e, r) \in \mathsf{rf}$. There are now two cases to consider: 1) $e \in E \setminus E^0$; or 2) $e \in E^0$.

In case (1), as G_k is PTSO-valid, we know that $(e, r) \in \mathsf{rf} \subseteq \mathsf{tso}{\cup}\mathsf{po}$. As such, from the construction of π we know that there exists π_3 such that $\pi_1 = \pi_3.\lambda.-$ and $\lambda{=}\mathsf{B}\langle e\rangle \vee \lambda{=}\mathsf{U}\langle e, -\rangle \vee (\lambda{=}\mathsf{W}\langle e\rangle \wedge \mathtt{tid}(e) = \mathtt{tid}(r))$. There are two more cases to consider: i) $\lambda{=}\mathsf{B}\langle e\rangle \vee \lambda{=}\mathsf{U}\langle e, -\rangle$; or ii) $\lambda{=}\mathsf{W}\langle e\rangle$.

In case (i) let us assume there exists e' such that $\mathtt{loc}(e'){=}\mathtt{loc}(r)$ and $\mathsf{B}\langle e'\rangle \in \pi_3$ or $\mathsf{U}\langle e', -\rangle \in \pi_3$. From the construction of π we then have $e' \in W \cup U$, $(e', r) \in \mathsf{tso}_t$ and $(e, e') \in \mathsf{tso}_t$. This however contradicts our result in (RF) and thus we have $\left\{\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi_3 \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset$, as required. Similarly, let us assume there exists e' such that $\mathtt{loc}(e'){=}\mathtt{loc}(r)$, $\mathtt{tid}(e') = \mathtt{tid}(r)$, $\mathsf{W}\langle e'\rangle \in \pi_3$ and $\mathsf{B}\langle e'\rangle \notin \pi_3$. From the construction of π we then have $e' \in W \cup U$, $(e', r) \in \mathsf{po}$ and $(e, e') \in \mathsf{po} \cap (W \cup U) \times (W \cup U) \subseteq \mathsf{tso}_t$. This however contradicts our result in (RF) and thus we have $\left\{e' \,\middle|\, \begin{array}{l}\mathsf{W}\langle e'\rangle \in \pi_3 \wedge \mathsf{B}\langle e'\rangle \notin \pi_3 \\ \mathtt{loc}(e'){=}\mathtt{loc}(r) \wedge \mathtt{tid}(e') = \mathtt{tid}(r)\end{array}\right\} = \emptyset$, as required. Similarly, in case (ii) we know that either $\mathsf{B}\langle e\rangle \in \pi_3$ or $\mathsf{B}\langle e\rangle \notin \pi_3$. In the former case the desired result follows from the proof of case (i). In the latter case, let us assume there exists e' such that $\mathtt{loc}(e'){=}\mathtt{loc}(r)$, $\mathtt{tid}(e') = \mathtt{tid}(r)$ and $\mathsf{W}\langle e'\rangle \in \pi_3$. From the construction of π we then have $e' \in W$, $(e', r) \in \mathsf{po}$ and $(e, e') \in \mathsf{po} \cap (W \cup U) \times (W \cup U) \subseteq \mathsf{tso}_t$. This however contradicts our result in (RF) and thus we have $\left\{\mathsf{W}\langle e'\rangle \in \pi_3 \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r) \wedge \mathtt{tid}(e') = \mathtt{tid}(r)\right\} = \emptyset$, as required.

In case (2), as G_k is PTSO-valid, we know either i) $k = 1 \wedge e = \mathit{init}_{\mathtt{loc}(e)}$; or ii) $k > 0 \wedge e = \max\left(G_{k-1}.\mathsf{nvo}|_{G_{k-1}.E^P \cap (U_{\mathtt{loc}(e)} \cup W_{\mathtt{loc}(e)})}\right)$. Let us now assume there exists e' such that $\mathsf{B}\langle e'\rangle \in \pi_1$ or $\mathsf{U}\langle e', -\rangle \in \pi_1$, and $\mathtt{loc}(e'){=}\mathtt{loc}(r)$. That is, $e' \in W \cup U$. From the construction of π we then have $(e', r) \in \mathsf{tso}_t$ and $(e, e') \in \mathsf{tso}_t$. This however contradicts our result in (RF) and thus we have $\left\{\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi_1 \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset$. Similarly, let us assume there exists e' such that $\mathtt{loc}(e'){=}\mathtt{loc}(r)$, $\mathtt{tid}(e') = \mathtt{tid}(r)$, $\mathsf{W}\langle e'\rangle \in \pi_1$. That is, $e' \in W \cup U$. From the construction of π we then have $(e', r) \in \mathsf{po}$ and $(e, e') \in \mathsf{po} \cap (W \cup U) \times (W \cup U) \subseteq \mathsf{tso}_t$. This however contradicts our result in (RF) and thus we have $\left\{\mathsf{W}\langle e'\rangle \in \pi_1 \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r) \wedge \mathtt{tid}(e') = \mathtt{tid}(r)\right\} = \emptyset$. In case (i), as $\Gamma_k = \epsilon$, we know $\pi'' = \epsilon$ and thus we simply have

$$\left\{\mathsf{PB}\langle e'\rangle \in \pi'' \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset$$

as required.

In case (ii), we then know either:

a) for all $b \in \{1 \cdots k{-}1\}$, $e \in G_b.E^0$ and $G_b.(W \cup U)_{\mathtt{loc}(e)} \setminus E^0 = \emptyset$ and thus $e = \mathit{init}_{\mathtt{loc}(e)}$; or

b) there exists $a \in \{1 \cdots k{-}1\}$ such that $e \in G_a.E^P \setminus E^0$, $\forall e' \in G_a.(W \cup U)_{\mathtt{loc}(e)}.\ (e', e) \in G_a.\mathsf{nvo}$ and for all $b \in \{a{+}1 \cdots k{-}1\}$, $e \in G_b.E^0$ and $G_b.(W \cup U)_{\mathtt{loc}(e)} \setminus E^0 = \emptyset$.

In case (a), let us assume there exists e' such that $\mathsf{PB}\langle e'\rangle \in \pi''$ and $\mathtt{loc}(e') = \mathtt{loc}(r) = \mathtt{loc}(e)$. We then know there exists $b \in \{1 \cdots k{-}1\}$ such that $e \in G_b.(W{\cup}U)_{\mathtt{loc}(e)} \setminus E^0$, leading to a contradiction. As such, we have

$$\left\{\mathsf{PB}\langle e'\rangle \in \pi'' \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset$$

as required.

In case (b), from the construction of $\pi_1 \cdots \pi_{k-1}$, we know there exists π_3, π_4 such that $\pi_a = \pi_3.\mathsf{PB}\langle e\rangle.\pi_4$, and $\pi'' = \pi_{k-1}.\cdots.\pi_a.\cdots.\pi_1$. Let us assume there exists e' such that $\mathsf{PB}\langle e'\rangle \in \pi_{k-1}.\cdots.\pi_{a+1}$ and $\mathtt{loc}(e') = \mathtt{loc}(r) = \mathtt{loc}(e)$. We then know either there exists $b \in \{k{-}1 \cdots a{+}1\}$ such that $e \in G_b.(W \cup U)_{\mathtt{loc}(e)} \setminus E^0$, leading to a contradiction. Similarly, let us assume there exists

e' such that $\mathsf{PB}\langle e' \rangle \in \pi_3$ and $\mathtt{loc}(e') = \mathtt{loc}(r) = \mathtt{loc}(e)$. We then know $(e, e') \in G_a.\mathsf{nvo}$, leading to a contradiction. As such, we have $\left\{\mathsf{PB}\langle e' \rangle \in \pi_{k-1}. \cdots .\pi_{a+1}.\pi_3 \,\middle|\, \mathtt{loc}(e'){=}\mathtt{loc}(r)\right\} = \emptyset$, as required.

For part (23), pick arbitrary e_1, e_2 such that $\mathtt{tid}(e_1) = \mathtt{tid}(e_2)$. For the $\Rightarrow$ direction assume $\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{W}\langle e_2 \rangle$. Moreover, from the construction of π we know that for all e such that $\mathtt{tid}(e) = \mathtt{tid}(e_1)$ we have $(e_1, e) \in \mathsf{po} \iff \mathsf{W}\langle e_1 \rangle <_\pi \mathtt{genL}(e, G_k)$. As such, we have $(e_1, e_2) \in \mathsf{po}$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{tso}$. Consequently, from the construction of π we have $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$, as required.

For the $\Leftarrow$ direction, assume $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$. From the construction of π we have $(e_1, e_2) \in \mathsf{tso}$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{po}$. Consequently, from the construction of π we have $\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{W}\langle e_2 \rangle$, as required.

For part (24), pick arbitrary e_1, e_2 such that $\mathtt{tid}(e_1) = \mathtt{tid}(e_2)$ and $\mathsf{W}\langle e_1 \rangle <_\pi \mathsf{F}\langle e_2 \rangle$. We then know there exists j such that $w_j = e_1$. Moreover, from the construction of π we know that for all e such that $\mathtt{tid}(e) = \mathtt{tid}(e_1)$ we have $(e_1, e) \in \mathsf{po} \iff \mathsf{W}\langle e_1 \rangle <_\pi \mathtt{genL}(e, G_k)$. As such, by definition we have $(e_1, e_2) \in \mathsf{po}$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{tso}$. Consequently, from the construction of π we have $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{F}\langle e_2 \rangle$, as required.

The proofs of parts (25), (26) and (27) are analogous and omitted here.

For part (28), pick arbitrary e_1, e_2 such that $\mathtt{loc}(e_1) = \mathtt{loc}(e_2)$. For the $\Rightarrow$ direction, assume $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$. From the construction of π we then know that $(e_1, e_2) \in \mathsf{tso}$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{nvo}$. Consequently, from the construction of π we have $\mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle$, as required.

For the $\Leftarrow$ direction, assume $\mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle$. From the construction of π we then know that $(e_1, e_2) \in \mathsf{nvo}$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{tso}$. Consequently, from the construction of π we have $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$, as required.

Similarly, for part (29), pick arbitrary $e_1, e_2 \in (PE \times PE) \setminus (W \cup U \times W \cup U)$. For the $\Rightarrow$ direction, assume $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$ or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$. From the construction of π we then know that $(e_1, e_2) \in \mathsf{tso}$. Moreover, we know that $(e_1, e_2) \in [W \cup U \cup PF]; \mathsf{tso}; [PF] \;\cup\; [PF]; \mathsf{tso}; [W \cup U \cup PF]$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{nvo}$. Consequently, from the construction of π we have $\mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle$, as required.

For the $\Leftarrow$ direction, assume $\mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle$. From the construction of π we then know that $(e_1, e_2) \in \mathsf{nvo}$. As $(e_1, e_2) \in [W \cup U \cup PF]; \mathsf{tso}; [PF] \;\cup\; [PF]; \mathsf{tso}; [W \cup U \cup PF]$ and G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{tso}$. Consequently, from the construction of π we have $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$ or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{B}\langle e_2 \rangle$ or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{U}\langle e_2, - \rangle$or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{PF}\langle e_2 \rangle$, as required.

For part (30), pick arbitrary e_1, e_2 such that $\mathsf{B}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle$ or $\mathsf{U}\langle e_1, - \rangle <_\pi \mathsf{PS}\langle e_2 \rangle$ or $\mathsf{PF}\langle e_1 \rangle <_\pi \mathsf{PS}\langle e_2 \rangle$. From the construction of π we then know that $(e_1, e_2) \in \mathsf{tso}$. Moreover, we know that $(e_1, e_2) \in \mathsf{tso}|_{PE}; [PS]$. As G_k is PTSO-valid, we then know that $(e_1, e_2) \in \mathsf{nvo}$. Consequently, from the construction of π we have $\mathsf{PB}\langle e_1 \rangle <_\pi \mathsf{PB}\langle e_2 \rangle$, as required.

Goal: complete$(\pi'_k.\pi_k)$
Follows immediately from the constructions of π'_k and π_k.

As $\mathsf{wfp}(\pi'_k.\pi_k, \mathtt{hist}(\Gamma_k))$ and $\mathsf{complete}(\pi'_k.\pi_k)$ hold, we know $\mathtt{getG}(\Gamma_k, \pi_k, \pi'_k)$ is defined. From the constructions of π'_k and π_k, it is now straightforward to demonstrate that $\mathtt{getG}(\Gamma_k, \pi_k, \pi'_k) = G_k$. □

Definition A.4. Given a $\Gamma = (G_n, (\pi'_n, \pi_n)). \cdots .(G_1, (\pi'_1, \pi_1))$ and an event path π, let

$$\mathsf{wf}(\Gamma, \pi) \stackrel{\text{def}}{\iff} \mathsf{wfp}(\pi, \mathcal{H}) \wedge \bigwedge_{i=1}^{n} \mathtt{getG}(\Gamma_i, \pi_i, \pi'_i) = G_i \wedge \mathsf{wfh}(\mathcal{H})$$

where $\Gamma_1{=}\epsilon$; $\Gamma_{i+1} = (G_i, (\pi'_i, \pi_i)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $i \in \{1 \cdots n{-}1\}$; and $\mathcal{H}{=}\mathtt{hist}(\Gamma)$.

Lemma A.3. *Let $\mathcal{E} = G_1; \cdots; G_n$ denote a PTSO-valid execution chain. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. For all $i \in \{1 \cdots n\}$:*

(1) for all $(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_i, S_i)$, and for all π, π':

$$\pi'_i.\pi_i = \pi'.\pi \Rightarrow \mathsf{wf}(\Gamma_i, \pi)$$

where $\Gamma_1{=}\epsilon$ and $\Gamma_{j+1}{=}(G_j, (\pi'_j, \pi_j)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $j \in \{1 \cdots i{-}1\}$.

(2) for all $(\pi'_n, \pi_n). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_n, S_n)$, $\pi'_n = \epsilon$.

Proof. Pick an arbitrary PTSO-valid execution chain $\mathcal{E} = G_1; \cdots; G_n$. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$.

RTS. (1) We proceed by induction on i.

Base case $i = 1$

Pick arbitrary $(\pi'_1, \pi_1) \in \mathtt{traces}(G_1, S_1)$ and π, π' such that $\pi'_1.\pi_1 = \pi'.\pi$. We are then required to show $\mathsf{wf}(\Gamma_1, \pi)$, where $\Gamma_1 = \epsilon$. It thus suffices to show:

$$\mathsf{wfp}(\pi, \mathtt{hist}(\Gamma_1)) \wedge \mathsf{wfh}(\mathtt{hist}(\Gamma_1))$$

The second conjunct follows trivially from the fact that $\mathtt{hist}(\Gamma_1) = \epsilon$ and the definition of $\mathsf{wfh}(\epsilon)$. As $(\pi'_1, \pi_1) \in \mathtt{traces}(G_1, S_1)$, from the definition of $\mathtt{traces}(.,.)$ we have $\mathtt{getG}(\Gamma_1, \pi_1, \pi'_1)$. Consequently, from the definition of $\mathtt{getG}(\Gamma_1, \pi_1, \pi'_1)$ we know that $\mathsf{wfp}(\pi'_1.\pi_1, \mathtt{hist}(\Gamma_1))$ holds implying the result in the first conjunct.

Base case $i = j{+}1$

$$\forall (\pi'_j, \pi_j). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_j, S_j).\ \forall \pi, \pi'.\ \pi'_j.\pi_j = \pi^2.\pi^1 \Rightarrow \mathsf{wf}(\Gamma'_j, \pi) \qquad \text{(I.H.)}$$

where $\Gamma'_1{=}\epsilon$ and $\Gamma'_{l+1}{=}(G_l, (\pi'_l, \pi_l)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $l \in \{1 \cdots j{-}1\}$.

Pick arbitrary $(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_i, S_i)$ and π, π' such that $\pi'_i.\pi_i = \pi'.\pi$. We are then required to show $\mathsf{wf}(\Gamma_i, \pi)$. It thus suffices to show:

$$\mathsf{wfp}(\pi, \mathtt{hist}(\Gamma_i)) \wedge \bigwedge_{k=1}^{j} \mathtt{getG}(\Gamma_k, \pi_k, \pi'_k) = G_k \wedge \mathsf{wfh}(\mathtt{hist}(\Gamma_i))$$

where $\Gamma_1{=}\epsilon$ and $\Gamma_{l+1}{=}(G_l, (\pi'_l, \pi_l)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $l \in \{1 \cdots j{-}1\}$.

The second conjunct follows from the definition of $\mathtt{traces}(.,.)$ and the fact that $(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_i, S_i)$. Similarly, as $(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \mathtt{traces}(G_i, S_i)$, from the definition of $\mathtt{traces}(.,.)$ we know $\mathtt{getG}(\Gamma_i, \pi_i, \pi'_i) = G_i$ and thus $\mathsf{wfp}(\pi'_i.\pi_i, \mathtt{hist}(\Gamma_i))$ holds implying the result in the first conjunct.

For the third conjunct, observe that $\texttt{hist}(\Gamma_i) = (\pi'_j, \pi_j).\texttt{hist}(\Gamma_j)$. As $(\pi'_i, \pi_i). \cdots .(\pi'_1, \pi_1) \in \texttt{traces}(G_i, S_i)$, from the definition of $\texttt{traces}(.,.)$ we know that $\texttt{getG}(\Gamma_j, \pi_j, \pi'_j) = G_j$ and thus $\mathsf{wfp}(\pi'_j.\pi_j, \texttt{hist}(\Gamma_j))$ and $\mathsf{complete}(\pi'_j.\pi_j)$ hold. On the other hand, from (I.H.) we have $\mathsf{wfh}(\texttt{hist}(\Gamma_j))$. As such, from the definition of $\mathsf{wfh}(.)$ we have $\mathsf{wfh}(\Gamma_i)$, as required.

RTS. (2) We proceed by contradiction. Assume there exists $(\pi'_n, \pi_n). \cdots .(\pi'_1, \pi_1) \in \texttt{traces}(G_n, S_n)$ such that $\pi'_n \neq \epsilon$. Let $\Gamma_1{=}\epsilon$ and $\Gamma_{j+1}{=}(G_j, (\pi'_j, \pi_j)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $j \in \{1 \cdots i{-}1\}$. From the definition of $\texttt{traces}(.,.)$ we then know that $\texttt{getG}(\Gamma_n, \pi_n, \pi'_n) = G_n$, i.e. $\mathsf{wfp}(\pi'_n.\pi_n, \texttt{hist}(\Gamma_n))$ and $\mathsf{complete}(\pi'_n.\pi_n)$ hold. As $\pi'_n \neq \epsilon$, we then know there exists $e \in G_n.E$ such that $\mathsf{PB}\langle e \rangle \in \pi'_n$, i.e. (from the well-formedness of the path) $\mathsf{PB}\langle e \rangle \notin \pi_n$. As such, since $\texttt{getG}(\Gamma_n, \pi_n, \pi'_n) = G_n$, from its definition we know that $e \notin G_n.E^P$. This however contradicts the assumption that G_n is PTSO-valid. □

Lemma A.4. *Let $\mathcal{E} = G_1; \cdots ; G_n$ denote a PTSO-valid execution chain of program* P *with outcome O and $G_i = (E^0_i, E^P_i, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$ for $i \in \{1, \cdots, n\}$. For each G_i, let $e^1_i, \cdots, e^m_i$ denote an enumeration of $E_i \setminus E^0_i$ that respects* po_i*. Then there exists $\mathsf{P}^1_i \cdots \mathsf{P}^m_i, \mathsf{S}^1_i, \mathsf{S}^m_i$ such that:*

- $\mathsf{P}^{j-1}_i, \mathsf{S}^{j-1}_i \; (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \xrightarrow{\texttt{genL}(e^j_i, G_i)} (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \; \mathsf{P}^j_i, \mathsf{S}^j_i$, *for $i \in \{1 \cdots n\}$ and $j \in \{1 \cdots m\}$*
- $\mathsf{P}^m_n = \textbf{skip} || \cdots || \textbf{skip}$ *and* $\mathsf{S}^m_n = O$

where $\mathsf{P}^0_1 = \mathsf{P}$; $\mathsf{P}^0_i = \textbf{recover}$ for $i \in \{2 \cdots n\}$; and $\mathsf{S}^0_i = \mathsf{S}_0$ for $i \in \{1 \cdots n\}$.

Lemma A.5. *Let $\mathcal{E} = G_1; \cdots ; G_n$ denote a PTSO-valid execution chain of program* P *with outcome O. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. Then, for all $i \in \{1 \cdots n\}$, and all $H_i. \cdots .H_1 \in \texttt{traces}(G_i, S_i)$:*

(1) if $i < n$ then
$\mathsf{P}^0_i, \mathsf{S}_0, \Gamma_i, \epsilon \Rightarrow^* \textbf{recover}, \mathsf{S}_0, \Gamma_{i+1}, \epsilon$

(2) $\mathsf{P}^0_n, \mathsf{S}_0, \Gamma_n, \epsilon \Rightarrow^* \textbf{skip} || \cdots || \textbf{skip}, O, \Gamma_n, \pi_n$

where $\mathsf{P}^0_1 = \mathsf{P}$; $\mathsf{P}^0_{j+1} = \textbf{recover}$; $\Gamma_1 = \epsilon$ and $\Gamma_{j+1}{=}(G_j, H_j). \cdots .(G_1, H_1)$, for $j \in \{1 \cdots n{-}1\}$.

Proof. Pick an arbitrary program P and a PTSO-valid execution chain $\mathcal{E}$ of P with outcome O such that $\mathcal{E} = G_1; \cdots ; G_n$. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. Let $\mathsf{P}^0_1 = \mathsf{P}$ and $\mathsf{P}^0_j = \textbf{recover}$ for $j \in \{2 \cdots n\}$ For all $i \in \{1 \cdots n\}$, pick arbitrary $(\pi'_i, \pi_i) \in \texttt{traces}(G_i, S_i)$. Let $\Gamma_1 = \epsilon$ and $\Gamma_{j+1} = (G_j, (\pi'_j, \pi_j)). \cdots .(G_1, (\pi'_1, \pi_1))$ for $j \in \{1 \cdots n\}$.

Part (1). Pick arbitrary $i < n$. From $\texttt{traces}(G_i, S_i)$ we know π_i respects $G_i.\mathsf{po}$. That is, π_i is of the form: $s_m.\texttt{genL}(e_m, G_i). \cdots .s_1.\texttt{genL}(e_1, G_i).s_0$, where:
i) For each $j \in \{0 \cdots m\}$, $s_j = \lambda_{(j,k_j)}. \cdots .\lambda_{(j,1)}$ and each $\lambda_{(j,r)}$ is either of the form $\mathsf{B}\langle - \rangle$ or of the form $\mathsf{PB}\langle - \rangle$, for $r \in \{1 \cdots k_j\}$; and
ii) $e_1 \cdots e_m$ denotes an enumeration of $G_i.E$ that respects $G_i.\mathsf{po}$ (for all e, e', if $(e, e') \in G_i.\mathsf{po}$ then $\texttt{genL}(e, G_i) <_{\pi_i} \texttt{genL}(e', G_i)$).

Moreover, since $(\pi'_i, \pi_i) \in \texttt{traces}(G_i, S_i)$, from the definition of $\texttt{traces}(.,.)$ we know that $\texttt{getG}(S_i, \pi_i, \pi'_i){=}G_i$. Additionally, from Lemma A.3 we know

$$\forall \lambda, p, q.\; \pi'_i.\pi_i = p.\lambda.q \Rightarrow \mathsf{fresh}(\lambda, p.q) \land \mathsf{fresh}(\lambda, \Gamma_i) \tag{31}$$

From (G-Prop) we thus have $\mathsf{P}^0_i, \mathsf{S}_0, \Gamma_i, \epsilon \Rightarrow^* \mathsf{P}^0_i, \mathsf{S}_0, \Gamma_i, s_0$. There are now two cases to consider: 1) $m = 0$; or 2) $m > 0$.

In case (1), we have $\pi_i = s_0$ and thus (since each event in s_0 is either of the form $\mathsf{B}\langle - \rangle$ or of the form $\mathsf{PB}\langle - \rangle$) from Lemma A.3 we know $s_0 = \pi_i = \pi'_i = \epsilon$. As such, we have $\mathsf{P}^0_i, \mathsf{S}_0, \Gamma_i, \epsilon \Rightarrow^*$

$\mathsf{P}_i^0, \mathsf{S}_0, \Gamma_i, \epsilon$. Moreover, since $\pi_i' = \epsilon$ then $\texttt{comp}(\pi_i, \pi_i')$ holds. As such from (G-Crash) we have $\mathsf{P}_i^0, \mathsf{S}_0, \Gamma_i, \epsilon \Rightarrow^* \textbf{recover}, \mathsf{S}_0, \Gamma_{i+1}, \epsilon$, as required.

In case (2) from Lemma A.4 we know there exists $\mathsf{P}_i^1 \cdots \mathsf{P}_i^m, \mathsf{S}_i^1, \mathsf{S}_i^m$ such that for $j \in \{1 \cdots m\}$:

$$\mathsf{P}_i^{j-1}, \mathsf{S}_i^{j-1} \ (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \xrightarrow{\texttt{genL}(e_i^j, G_i)} (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \ \mathsf{P}_i^j, \mathsf{S}_i^j \tag{32}$$

where $\mathsf{S}_i^0 = \mathsf{S}_0$ for $i \in \{1 \cdots n\}$.

For each $j \in \{1 \cdots m\}$, from (32) we then know there exist $\mathsf{P}_j', \mathsf{P}_j'', \mathsf{S}_j', \mathsf{S}_j''$ such that $\mathsf{P}_i^{j-1}, \mathsf{S}_i^{j-1} (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \mathsf{P}_j', \mathsf{S}_j' \xrightarrow{\texttt{genL}(e_i^j, G_i)} \mathsf{P}_j'', \mathsf{S}_j'' (\xrightarrow{\mathcal{E}\langle\tau\rangle})^* \ \mathsf{P}_i^j, \mathsf{S}_i^j$. Let $p_0 = s_0$ and $p_j = s_j.\texttt{genL}(e_j, G_i). \cdots .s_1.\texttt{genL}(e_1, G_i).s_0$, for $j \in \{1 \cdots m\}$. As such, from (G-SilentP), (G-Step), (G-Prop), and (31) we then have:

$$\begin{array}{ll} & \mathsf{P}_i^{j-1}, \mathsf{S}_i^{j-1}, \Gamma_i, p_{j-1} \\ \Rightarrow^* & \mathsf{P}_j', \mathsf{S}_j', \Gamma_i, p_{j-1} \\ \Rightarrow & \mathsf{P}_j'', \mathsf{S}_j'', \Gamma_i, \texttt{genL}(e_j, G_i).p_{j-1} \\ \Rightarrow^* & \mathsf{P}_i^j, \mathsf{S}_i^j, \Gamma_i, \texttt{genL}(e_j, G_i).p_{j-1} \\ \Rightarrow & \mathsf{P}_i^j, \mathsf{S}_i^j, \Gamma_i, p_j \end{array}$$

Consequently, we have

$$\mathsf{P}_i^0, \mathsf{S}_0, \Gamma_i, \epsilon \Rightarrow^* \mathsf{P}_i^0, \mathsf{S}_i^0, \Gamma_i, p_0 \Rightarrow^* \mathsf{P}_i^1, \mathsf{S}_i^1, \Gamma_i, p_1 \Rightarrow^* \cdots \Rightarrow^* \mathsf{P}_i^m, \mathsf{S}_i^m, \Gamma_i, p_m$$

That is, we have

$$\mathsf{P}_i^0, \mathsf{S}_i^0, \Gamma_i, \epsilon \Rightarrow^* \mathsf{P}_i^m, \mathsf{S}_i^m, \Gamma_i, \pi_i$$

On the other hand from Lemma A.3 we know that $\texttt{comp}(\pi, \pi')$ holds. As such, since $\texttt{getG}(S_i, \pi_i, \pi_i')$=$G_i$, from (G-Crash) we have

$$\mathsf{P}_i^m, \mathsf{S}_i^m, \Gamma_i, \pi_i \Rightarrow^* \textbf{recover}, \mathsf{S}_0, \Gamma_{i+1}, \epsilon$$

That is, we have $\mathsf{P}_i^0, \mathsf{S}_i^0, \Gamma_i, \epsilon \Rightarrow^* \textbf{recover}, \mathsf{S}_0, \Gamma_{i+1}, \epsilon$, as required.

Part (2). From $\texttt{traces}(G_n, S_n)$ we know π_n respects $G_n.\text{po}$. That is, π_n is of form: $s_m.\texttt{genL}(e_m, G_n) . \cdots .s_1.\texttt{genL}(e_1, G_n).s_0$, where:
i) For each $j \in \{0 \cdots m\}$, $s_j = \lambda_{(j,k_j)}. \cdots .\lambda_{(j,1)}$ and each $\lambda_{(j,r)}$ is either of the form $\mathsf{B}\langle - \rangle$ or of the form $\mathsf{PB}\langle - \rangle$, for $r \in \{1 \cdots k_j\}$; and
ii) $e_1 \cdots e_m$ denotes an enumeration of $G_n.E$ that respects $G_i.\text{po}$ (for all e, e', if $(e, e') \in G_n.\text{po}$ then $\texttt{genL}(e, G_n) <_{\pi_n} \texttt{genL}(e', G_n)$).

Moreover, since $(\pi_n', \pi_n) \in \texttt{traces}(G_n, S_n)$, from the definition of $\texttt{traces}(.,.)$ we know that $\texttt{getG}(S_n, \pi_n, \pi_n')$=$G_n$. Additionally, from Lemma A.3 we know:

$$\pi_n' = \epsilon \land \forall \lambda, p, q.\ \pi_n'.\pi_n = p.\lambda.q \Rightarrow \mathsf{fresh}(\lambda, p.q) \land \mathsf{fresh}(\lambda, \Gamma_n) \tag{33}$$

From (G-Prop) we thus have $\mathsf{P}_n^0, \mathsf{S}_0, \Gamma_n, \epsilon \Rightarrow^* \mathsf{P}_n^0, \mathsf{S}_0, \Gamma_n, s_0$. There are now two cases to consider: 1) $m = 0$; or 2) $m > 0$.

In case (1), we have $\mathsf{P}_n^0 = \textbf{skip} || \cdots || \textbf{skip}$, $\mathsf{S}_n^0 = \mathsf{S}_0 = O$, and $\pi_n = s_0$ and thus (since each event in s_0 is either of the form $\mathsf{B}\langle - \rangle$ or of the form $\mathsf{PB}\langle - \rangle$) from Lemma A.3 we know $s_0 = \pi_n = \pi_n' = \epsilon$. As such, we trivially have $\mathsf{P}_n^0, \mathsf{S}_0, \Gamma_n, \epsilon \Rightarrow^* \textbf{skip} || \cdots || \textbf{skip}, O, \Gamma_n, \epsilon$, as required.

In case (2), in similar steps to that of the proof of part (1) we have:

$$\mathsf{P}_n^0, \mathsf{S}_n^0, \Gamma_n, \epsilon \Rightarrow^* \mathsf{P}_n^m, \mathsf{S}_n^m, \Gamma_n, \pi_n$$

That is, we have $\mathsf{P}_n^0, \mathsf{S}_n^0, \Gamma_n, \epsilon \Rightarrow^* \textbf{skip} || \cdots || \textbf{skip}, O, \Gamma_n, \pi_n$, as required.

□

Corollary 1. *Let $\mathcal{E} = G_1; \cdots ; G_n$ denote a PTSO-valid execution chain of program* P *with outcome O. Let $S_1 = \epsilon$ and $S_{j+1} = G_j. \cdots .G_1$ for $j \in \{1 \cdots n\}$. Then, there exists $H_n. \cdots .H_1 \in \texttt{traces}(G_n, S_n)$, with $H_n = (-, \pi_n)$ such that*

$$\mathsf{P}, \mathsf{S}_0, \epsilon, \epsilon \Rightarrow^* \textbf{skip} || \cdots || \textbf{skip}, O, (G_{n-1}, H_{n-1}). \cdots .(G_1, H_1), \pi_n$$

Proof. Follows from Lemma A.2 and Lemma A.5. □

Given an execution path π and a graph history Γ, the set of configurations induced by Γ and π, written $\texttt{confs}(\Gamma, \pi)$, includes those configurations that satisfy the following condition:

$$\texttt{confs}(\Gamma, \pi) \triangleq \left\{(M, PB, B) \,\middle|\, \mathsf{wf}(M, PB, B, \texttt{hist}(\Gamma), \pi)\right\}$$

Lemma A.6. *For all* $\mathsf{P}, \mathsf{P}', \mathsf{S}, \mathsf{S}', \Gamma, \Gamma', \pi, \pi'$:
if

$$\mathsf{wf}(\Gamma, \pi) \wedge \mathsf{wf}(\Gamma', \pi') \wedge \mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \mathsf{P}', \mathsf{S}', \Gamma', \pi'$$

then for all $(M, PB, B) \in \texttt{confs}(\Gamma, \pi)$*, there exists* $(M', PB', B) \in \texttt{confs}(\Gamma', \pi')$ *such that*

$$\mathsf{P}, \mathsf{S}, M, PB, B, \texttt{hist}(\Gamma), \pi \Rightarrow^* \mathsf{P}', \mathsf{S}', M', PB', B', \texttt{hist}(\Gamma'), \pi'$$

Proof. Pick arbitrary $\mathsf{P}, \mathsf{P}', \mathsf{S}, \mathsf{S}', \Gamma, \Gamma', \pi, \pi'$ such that $\mathsf{wf}(\Gamma, \pi)$, $\mathsf{wf}(\Gamma', \pi')$, and $\mathsf{P}, \mathsf{S}, \Gamma, \pi \Rightarrow \mathsf{P}', \mathsf{S}', \Gamma', \pi'$. Pick arbitrary $(M, PB, B) \in \texttt{confs}(\Gamma, \pi)$. Let $\mathcal{H} = \texttt{hist}(\Gamma)$. From the definition of $\texttt{confs}(.,.)$ we then know that $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds. We then proceed by induction on the structure of $\Rightarrow$.

Case (G-SilentP)
From (G-SilentP) we then know that $\mathsf{P}, \mathsf{S} \xrightarrow{\mathcal{E}\langle\tau\rangle} \mathsf{P}', \mathsf{S}'$, and that $\Gamma' = \Gamma$, $\pi' = \pi$. As such, from (A-SilentP) we have $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}', \mathsf{S}', M, PB, B, \mathcal{H}, \pi$. Moreover, as $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, the required result holds immediately.

Case (G-Prop)
From (G-Prop) we then know that there exists e and $\lambda \in \left\{\mathsf{B}\langle e\rangle, \mathsf{PB}\langle e\rangle\right\}$ such that $\pi' = \lambda.\pi$, $\mathsf{fresh}(\lambda, \pi)$, $\mathsf{fresh}(\lambda, \Gamma)$, $\mathsf{P}' = \mathsf{P}$, $\mathsf{S}' = \mathsf{S}$ and $\Gamma' = \Gamma$. From the definition of $\mathsf{fresh}(.,.)$ we then know that $\mathsf{fresh}(\lambda, \mathcal{H})$ holds. There are now three cases to consider. Either 1) $\lambda = \mathsf{B}\langle e\rangle$; or 2) $\lambda = \mathsf{PB}\langle e\rangle$ and $e \in W \cup U$; or 3) $\lambda = \mathsf{PB}\langle e\rangle$ and $e \in PF$.

In case (1), let $\texttt{tid}(e) = \tau$, $\texttt{loc}(e) = \texttt{x}$. Since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know there exist pb'', PB such that $PB = (\textsc{None}, pb).PB''$. In what follows, we demonstrate that there exists b such that $B(\tau) = b.e$. From (AM-BProp) we then have $M, PB, B \xrightarrow{\mathsf{B}\langle e\rangle} M, (\textsc{None}, pb[\texttt{x} \mapsto e.PB(\texttt{x})]).PB'', B[\tau \mapsto b]$. As such, from (A-PropM) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, (\textsc{None}, pb[\texttt{x} \mapsto e.PB(\texttt{x})]).PB', B[\tau \mapsto b], \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = (\textsc{None}, pb[\texttt{x} \mapsto e.pb''(\texttt{x})]).PB''$ and $B' = B[\tau \mapsto b]$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required. We next demonstrate that there exists b such that $B(\tau) = b.e$.

Since $\mathsf{wf}(\Gamma', \pi')$ holds, we know that $\mathsf{W}\langle e\rangle \in \pi$. Moreover, as $\mathsf{fresh}(\lambda, \pi)$, we know that $\lambda \notin \pi$. As such, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know that $e \in B(\tau)$. Now let us suppose that e is not at the head of $B(\tau)$, i.e. there exists $e' \neq e$ and b such that $e' \prec_{B(\tau)} e$. Once again, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know that $\mathsf{W}\langle e' \rangle \in \pi$, $\mathsf{B}\langle e' \rangle \notin \pi$ (and thus $\mathsf{B}\langle e' \rangle \notin \lambda.\pi$) and that $\mathsf{W}\langle e' \rangle \prec_\pi \mathsf{W}\langle e \rangle$. Moreover, since $alb \in \lambda.\pi$ and $\mathsf{wf}(\Gamma, \lambda.\pi)$ holds, from the definition of $\mathsf{wf}(.,.)$ and the definition of $\mathsf{wfp}(.,.)$ we know that $\mathsf{B}\langle e' \rangle \prec_{\lambda.\pi} \mathsf{B}\langle e \rangle$. This however leads to a contradiction as $\mathsf{B}\langle e' \rangle \notin \lambda.\pi$. We can thus conclude that there exists b such that $B(\tau) = b.e$.

In case (2), let $PB = PB''.(o, pb)$ and let $\mathtt{loc}(e) = \mathtt{x}$. In what follows, we demonstrate that there exists s such that $pb(\mathtt{x}) = s.e$. From (AM-PBProp) we then have $M, PB, B \xrightarrow{\mathsf{PB}\langle e \rangle} M[\mathtt{x} \mapsto e], PB''.(\textsc{None}, pb[\mathtt{x} \mapsto s]), B$. As such, from (A-PropM) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M[\mathtt{x} \mapsto e], PB''.(o, pb[\mathtt{x} \mapsto s]), B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M[\mathtt{x} \mapsto e]$, $PB' = PB''.(o, pb[\mathtt{x} \mapsto s])$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\mathtt{confs}(.,.)$ we have $(M', PB', B') \in \mathtt{confs}(\Gamma, \pi')$, as required. We next demonstrate that there exists s such that $pb(\mathtt{x}) = s.e$.

Since $\mathsf{wf}(\Gamma', \pi')$ holds, we know that there exists $\lambda_e \in \pi$ such that $\lambda_e = \mathsf{U}\langle e, - \rangle$ or $\lambda_e = \mathsf{B}\langle e \rangle -$. Moreover, as $\mathsf{fresh}(\lambda, \pi)$, we know that $\lambda \notin \pi$. As such, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know there exists $(o_e, pb_e) \in PB$ such that $e \in pb_e(\mathtt{x})$. Now let us suppose that e is not the next event in PB to be propagated, i.e. either i) there exists $(o_{e'}, pb_{e'}) \in PB$ such that $(o_{e'}, pb_{e'}) \prec_{PB} (o_e, pb_e)$ and either $o_{e'} = \textsc{Some}(e')$ or there exists $\mathtt{y}$ such that $e' \in pb_{e'}(\mathtt{y})$; or ii) $e' \prec_{pb_e(\mathtt{x})} e$. Once again, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know that there exists $\lambda_{e'} \in \pi$ such that $\lambda_{e'} = \mathsf{B}\langle e' \rangle$, or $\lambda_{e'} = \mathsf{U}\langle e', - \rangle$ or $\lambda_{e'} = \mathsf{PF}\langle e' \rangle$, that $\mathsf{PB}\langle e' \rangle \notin \pi$ (and thus $\mathsf{PB}\langle e' \rangle \notin \lambda.\pi$) and that $\lambda_{e'} \prec_\pi \lambda_e$. Moreover, since $\lambda \in \lambda.\pi$ and $\mathsf{wf}(\Gamma, \lambda.\pi)$ holds, from the definition of $\mathsf{wf}(.,.)$ and the definition of $\mathsf{wfp}(.,.)$ we know that $\mathsf{PB}\langle e' \rangle \prec_{\lambda.\pi} \mathsf{PB}\langle e \rangle$. This however leads to a contradiction as $\mathsf{PB}\langle e' \rangle \notin \lambda.\pi$. We can thus conclude that there exists s such that $pb(\mathtt{x}) = s.e$.

In case (3), let $PB = PB''.(o, pb)$. In what follows, we demonstrate that $(o, pb) = (\textsc{Some}(e), pb_0)$. From (AM-PBPropF) we then have $M, PB, B \xrightarrow{\mathsf{PB}\langle e \rangle} M, PB''.(\textsc{None}, pb_0), B$. As such, from (A-PropM) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, MPB''.(\textsc{None}, pb_0), B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = PB''.(\textsc{None}, pb_0)$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\mathtt{confs}(.,.)$ we have $(M', PB', B') \in \mathtt{confs}(\Gamma, \pi')$, as required. We next demonstrate that $(o, pb) = (\textsc{Some}(e), pb_0)$.

Since $\mathsf{wf}(\Gamma', \pi')$ holds, we know $\mathsf{PF}\langle e \rangle \in \pi$. Moreover, as $\mathsf{fresh}(\lambda, \pi)$, we know that $\lambda \notin \pi$. As such, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know there exists $(o_e, pb_e) \in PB$ such that $o_e = \textsc{Some}(e)$. Now let us suppose that e is not the next event in PB to be propagated, i.e. either i) there exists $(o_{e'}, pb_{e'}) \in PB$ such that $(o_{e'}, pb_{e'}) \prec_{PB} (o_e, pb_e)$ and either $o_{e'} = \textsc{Some}(e')$ or there exists $\mathtt{y}$ such that $e' \in pb_{e'}(\mathtt{y})$; or ii) there exists $\mathtt{y}$ such that $e' \in pb_e(\mathtt{y})$. Once again, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know that there exists $\lambda_{e'} \in \pi$ such that $\lambda_{e'} = \mathsf{B}\langle e' \rangle$, or $\lambda_{e'} = \mathsf{U}\langle e', - \rangle$ or $\lambda_{e'} = \mathsf{PF}\langle e' \rangle$, that $\mathsf{PB}\langle e' \rangle \notin \pi$ (and thus $\mathsf{PB}\langle e' \rangle \notin \lambda.\pi$) and that $\lambda_{e'} \prec_\pi \mathsf{PF}\langle e \rangle$. Moreover, since $\lambda \in \lambda.\pi$ and $\mathsf{wf}(\Gamma, \lambda.\pi)$ holds, from the definition of $\mathsf{wf}(.,.)$ and the definition of $\mathsf{wfp}(.,.)$ we know that $\mathsf{PB}\langle e' \rangle \prec_{\lambda.\pi} \mathsf{PB}\langle e \rangle$. This however leads to a contradiction as $\mathsf{PB}\langle e' \rangle \notin \lambda.\pi$. We can thus conclude that $(o, pb) = (\textsc{Some}(e), pb_0)$.

Case (G-Step)

We know there exists e, r, u and $\lambda \in \{\mathsf{R}\langle r, e \rangle, \mathsf{W}\langle e \rangle, \mathsf{U}\langle u, e \rangle, \mathsf{F}\langle e \rangle, \mathsf{PF}\langle e \rangle, \mathsf{PS}\langle e \rangle\}$ such that $\pi' = \lambda.\pi$, $\mathsf{fresh}(\lambda, \pi)$, $\mathsf{fresh}(\lambda, \Gamma)$, $\Gamma' = \Gamma$ and $\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}', \mathsf{S}'$. From the definition of $\mathsf{fresh}(.,.)$ we then know that fresh$(\lambda, \mathcal{H})$ holds. There are now six cases to consider. Either 1) $\lambda = \mathsf{R}\langle e, w\rangle$; or 2) $\lambda = \mathsf{W}\langle e\rangle$; or 3) $\lambda = \mathsf{U}\langle e, w\rangle$; or 4) $\lambda = \mathsf{F}\langle e\rangle$; or 5) $\lambda = \mathsf{PF}\langle e\rangle$; or 6) $\lambda = \mathsf{PS}\langle e\rangle$.

Case 1: $\lambda = \mathsf{R}\langle r, e\rangle$
Let $\mathtt{tid}(r) = \tau$, $\mathtt{loc}(r) = \mathtt{x}$ and $B(\tau) = b$. In what follows we demonstrate that $\mathtt{read}(M, PB, b, \mathtt{x}) = e$. From (AM-Read) we then have $M, PB, B \xrightarrow{\mathsf{R}\langle r,e\rangle} M, PB, B$. As such, from (A-Step) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = PB$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since wf$(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have wf$(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\mathtt{confs}(.,.)$ we have $(M', PB', B') \in \mathtt{confs}(\Gamma, \pi')$, as required. We next demonstrate that $\mathtt{read}(M, PB, b, \mathtt{x}) = e$.

From the definition of wf$(\Gamma, \lambda.\pi)$ we know that wfrd(r, e, π, π_h), where $\pi_h = \pi_n.\cdots.\pi_1$, when $\Gamma = (-, (\pi_n, -)).\cdots.(-, (\pi_1, -))$. From the definition of wfrd(r, e, π, π_h) there are now four cases to consider:

i) $\exists \pi_1, \pi_2.\ \pi = \pi_1.\mathsf{W}\langle e\rangle.\pi_2 \land \mathtt{tid}(e) = \mathtt{tid}(r) \land \mathsf{B}\langle e\rangle \notin \pi_1$
$\land \left\{\mathsf{W}\langle e'\rangle \in \pi_1 \;\middle|\; \mathtt{loc}(e') = \mathtt{loc}(r) \land \mathtt{tid}(e') = \mathtt{tid}(r)\right\} = \emptyset$

ii) $\exists \pi_1, \pi_2, \lambda_e.\ \pi = \pi_1.\lambda_e.\pi_2 \land (\lambda_e = \mathsf{B}\langle e\rangle \lor \lambda_e = \mathsf{U}\langle e, -\rangle)$
$\land \left\{\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi_1 \;\middle|\; \mathtt{loc}(e') = \mathtt{loc}(r)\right\} = \emptyset$
$\land \left\{e' \;\middle|\; \begin{array}{l}\mathsf{W}\langle e'\rangle \in \pi \land \mathsf{B}\langle e'\rangle \notin \pi \\ \land\, \mathtt{loc}(e') = \mathtt{loc}(r) \land \mathtt{tid}(e') = \mathtt{tid}(r)\end{array}\right\} = \emptyset$

iii) $\exists \pi_1, \pi_2.\ \pi_h = \pi_1.\mathsf{PB}\langle e\rangle.\pi_2$
$\land \left\{\begin{array}{l}\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi, \\ \mathsf{W}\langle e''\rangle \in \pi, \\ \mathsf{PB}\langle e'\rangle \in \pi_1\end{array} \;\middle|\; \begin{array}{l}\mathtt{loc}(e') = \mathtt{loc}(r) \land \\ \mathtt{loc}(e'') = \mathtt{loc}(r) \land \\ \mathtt{tid}(e'') = \mathtt{tid}(r)\end{array}\right\} = \emptyset$

iv) $e = init_{\mathtt{x}} \land \left\{\begin{array}{l}\mathsf{B}\langle e'\rangle, \mathsf{U}\langle e', -\rangle \in \pi, \\ \mathsf{W}\langle e''\rangle \in \pi, \\ \mathsf{PB}\langle e'\rangle \in \pi_h\end{array} \;\middle|\; \begin{array}{l}\mathtt{loc}(e') = \mathtt{loc}(r) \land \\ \mathtt{loc}(e'') = \mathtt{loc}(r) \land \\ \mathtt{tid}(e'') = \mathtt{tid}(r)\end{array}\right\} = \emptyset$

In case (i), since wf$(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know there exists b' such that $b = e.b'$. As such, by definition we have $\mathtt{read}(M, PB, b, \mathtt{x}) = e$.

In case (ii), since wf$(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know that for all $e' \in b$, $\mathtt{loc}(e') \neq \mathtt{x}$; and that there exists $PB_1, PB_2, (o, pb), s$ such that $PB = PB_1.(o, pb).PB_2$, $PB(\mathtt{x}) = e.s$ and for all $(o', pb') \in PB_1$, $pb'(\mathtt{x}) = \epsilon$. As such, by definition we have $\mathtt{read}(M, PB, b, \mathtt{x}) = e$.

In case (iii), since wf$(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know that for all $e' \in b$, $\mathtt{loc}(e') \neq \mathtt{x}$; that for all $(o, pb) \in PB$, $PB(\mathtt{x}) = \epsilon$; and that $M(\mathtt{x}) = e$. As such, by definition we have $\mathtt{read}(M, PB, b, \mathtt{x}) = e$.

In case (iv), , since wf$(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know that for all $e' \in b$, $\mathtt{loc}(e') \neq \mathtt{x}$; that for all $(o, pb) \in PB$, $PB(\mathtt{x}) = \epsilon$; and that $M(\mathtt{x}) = init_{\mathtt{x}}$. As such, by definition we have $\mathtt{read}(M, PB, b, \mathtt{x}) = e$.

Case 2: $\lambda = \mathsf{W}\langle e\rangle$
Let $\mathtt{tid}(e) = \tau$. From (AM-Write) we then have $M, PB, B \xrightarrow{\mathsf{W}\langle e\rangle} M, PB, B[\tau \mapsto e.B(\tau)]$. As such, from (A-Step) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, PB, B[\tau \mapsto e.B(\tau)], \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = PB$ and $B' = B[\tau \mapsto e.B(\tau)]$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required.

Case 3: $\lambda = \mathsf{U}\langle u, e\rangle$
Let $\texttt{tid}(u) = \tau$ and $\texttt{loc}(u) = \texttt{x}$. In what follows we demonstrate that $B(\tau) = \epsilon$. Since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know there exist pb'', PB such that $PB = (\textsc{None}, pb).PB''$. Moreover, in an analogous way to that in case (2) we can demonstrate that $\texttt{read}(M, PB, b, \texttt{x}) = e$. From (AM-RMW) we then have $M, PB, B \xrightarrow{\mathsf{U}\langle u, e\rangle} M, (\textsc{None}, pb[\texttt{x} \mapsto u.pb(\texttt{x})]).PB'', B$. As such, from (A-Step) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, (\textsc{None}, pb[\texttt{x} \mapsto u.pb(\texttt{x})]).PB'', B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = (\textsc{None}, pb[\texttt{x} \mapsto u.pb(\texttt{x})]).PB''$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required. We next demonstrate that $B(\tau) = \epsilon$.

Let us suppose that there exists e' such that $e' \in b(\tau)$. We then know that $\texttt{tid}(e') = \tau$. From the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we then know that $\mathsf{W}\langle e'\rangle \in \pi$, $\mathsf{B}\langle e'\rangle \notin \pi$ and thus $\mathsf{B}\langle e'\rangle \notin \lambda.\pi$. That is, we have $\mathsf{W}\langle e'\rangle \prec_{\lambda.\pi} \lambda$. Moreover, since $alb \in \lambda.\pi$ and $\mathsf{wf}(\Gamma, \lambda.\pi)$ holds, from the definition of $\mathsf{wf}(.,.)$ and the definition of $\mathsf{wfp}(.,.)$ we know that $\mathsf{B}\langle e'\rangle \prec_{\lambda.\pi} \mathsf{F}\langle e\rangle$. This however leads to a contradiction as $\mathsf{B}\langle e'\rangle \notin \lambda.\pi$. We can thus conclude that $B(\tau) = \epsilon$.

Case 4: $\lambda = \mathsf{F}\langle e\rangle$
Let $\texttt{tid}(e) = \tau$. In an analogous way to that in case (3) we can demonstrate that $B(\tau) = \epsilon$. From (AM-Fence) we then have $M, PB, B \xrightarrow{\mathsf{F}\langle e\rangle} M, PB, B$. As such, from (A-Step) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = PB$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required.

Case 5: $\lambda = \mathsf{PF}\langle e\rangle$
Let $\texttt{tid}(e) = \tau$. In an analogous way to that in case (3) we can demonstrate that $B(\tau) = \epsilon$. On the other hand, from $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ and the definition of $\texttt{pbuff}(.,.)$ in particular, we know that there exists pb and PB'' such that $PB = (\textsc{None}, pb).PB''$. As such, from (AM-PFence) we have: $M, PB, B \xrightarrow{\mathsf{PF}\langle e\rangle} M, (\textsc{None}, pb_0).(\textsc{Some}(e), pb).PB'', B$. As such, from (A-Step) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, (\textsc{None}, pb_0).(\textsc{Some}(e), pb).PB'', B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M' = M$, $PB' = (\textsc{None}, pb_0).(\textsc{Some}(e), pb).PB''$ and $B' = B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required.

Case 6: $\lambda = \mathsf{PS}\langle e\rangle$
Let $\texttt{tid}(e) = \tau$. In an analogous way to that in case (3) we can demonstrate that $B(\tau) = \epsilon$. In what follows we demonstrate that $PB = PB_0$. As such, from (AM-PSync) we have: $M, PB, B \xrightarrow{\mathsf{PS}\langle e\rangle} M, PB, B$.

As such, from (A-STEP) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \lambda.\pi$$

That is, there exists $M'{=}M$, $PB'{=}PB$ and $B'{=}B$ such that $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}, \pi'$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we also have $\mathsf{wf}(M', PB', B', \mathcal{H}, \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required. We next demonstrate that $PB = PB_0$.

Let us suppose $PB \neq PB_0$, i.e. there exist e' and $(o_{e'}, pb_{e'}) \in PB$ such that either i) $o_{e'} = \textsc{Some}(e')$; or ii) there exists y such that $e' \in pb_{e'}(\mathrm{y})$. Once again, from the definition of $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ we know that there exists $\lambda_{e'} \in \pi$ such that $\lambda_{e'} = \mathsf{B}\langle e'\rangle$, or $\lambda_{e'} = \mathsf{U}\langle e', -\rangle$ or $\lambda_{e'} = \mathsf{PF}\langle e'\rangle$, that $\mathsf{PB}\langle e'\rangle \notin \pi$ (and thus $\mathsf{PB}\langle e'\rangle \notin \lambda.\pi$) and that $\lambda_{e'} <_{\lambda.\pi} \lambda$. Moreover, since $\lambda \in \lambda.\pi$ and $\mathsf{wf}(\Gamma, \lambda.\pi)$ holds, from the definition of $\mathsf{wf}(.,.)$ and the definition of $\mathsf{wfp}(.,.)$ we know that $\mathsf{PB}\langle e'\rangle <_{\lambda.\pi} \mathsf{PS}\langle e\rangle$. This however leads to a contradiction as $\mathsf{PB}\langle e'\rangle \notin \lambda.\pi$. We can thus conclude that$PB = PB_0$.

Case (G-CRASH)
Let $\Gamma = (G_n, -). \cdots .(G_1, -)$. From (G-CRASH) we know there exists π'' and G such that $\mathsf{P}' = \textbf{recover}$, $\mathsf{S}' = \mathsf{S}_0$, $\Gamma' = (G, (\pi'', \pi)).\Gamma$, $\pi' = \epsilon$, $\texttt{comp}(\pi, \pi'')$ and $\texttt{getG}(G_n. \cdots .G_1, \pi, \pi'') = G$. since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds, from its definition we know that for all events e and all $(o, pb) \in PB$:
i) $e \in B(\texttt{tid}(e)) \iff \mathsf{W}\langle e\rangle \in \pi \land \mathsf{B}\langle e\rangle \notin \pi$; and that
ii) $e \in pb(\texttt{loc}(e)) \lor o = \textsc{Some}(e) \iff \mathsf{PB}\langle e\rangle \notin \pi \land (\mathsf{B}\langle e\rangle \in \pi \lor \mathsf{U}\langle e, -\rangle \in \pi \lor \mathsf{PF}\langle e\rangle \in \pi)$.
As such, from the definition of $\texttt{comp}(.,.)$ we know for all events e and all $(o, pb) \in PB$:
i) $e \in B(\texttt{tid}(e)) \iff \mathsf{B}\langle e\rangle \in \pi''$;
ii) $e \in pb(\texttt{loc}(e)) \lor o = \textsc{Some}(e) \iff \mathsf{PB}\langle e\rangle \in \pi''$.
As such, from the definition of $\rightarrow_{\mathrm{p}}$ we have $M, PB, B \xrightarrow{\pi''}_{\mathrm{p}} -, PB_0, B_0$. Consequently, from (A-STEP) we have:

$$\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}', \mathsf{S}', M, PB_0, B_0, (\pi'', \pi).\mathcal{H}, \pi'$$

That is, there exists $M' = M$, $PB' = PB_0$, $B' = B_0$ and $\mathcal{H}' = (\pi'', \pi).\mathcal{H} = \texttt{hist}(\Gamma')$ such that: $\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi \Rightarrow \mathsf{P}, \mathsf{S}, M', PB', B', \mathcal{H}', \pi'$. Since $\texttt{comp}(\pi, \pi'')$ holds, by definition we have $\mathsf{complete}(\pi''.\pi)$. Moreover, since $\mathsf{wf}(M, PB, B, \mathcal{H}, \pi)$ holds and $\mathsf{wf}(\Gamma', \pi')$ holds, from their definitions we also have $\mathsf{wf}(M', PB', B', \mathcal{H}', \pi')$ and thus from the definition of $\texttt{confs}(.,.)$ we have $(M', PB', B') \in \texttt{confs}(\Gamma, \pi')$, as required.

□

Theorem 5 (Completeness). *Given a program* P*, for all PTSO-valid execution chains* $\mathcal{E}$ *of* P *with outcome* O*, there exists* M*,* $\mathcal{H}$ *and* π *such that*

$$\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \textbf{skip}|| \cdots ||\textbf{skip}, O, M, PB_0, B_0, \mathcal{H}, \pi$$

PROOF. Follows from Corollary 1, Lemma A.3 and Lemma A.6. □

A.4 Equivalence of PTSO Operational and Intermediate Semantics

Let

$$R_l \triangleq \left\{((\tau : \mathsf{l}), \lambda) \;\middle|\; \begin{array}{l} (\exists e.\ \texttt{getE}(\lambda) = e \land \texttt{tid}(e) = \tau \land \texttt{lab}(e) = \mathsf{l}) \lor (\lambda = \mathcal{E}\langle\tau\rangle \land \mathsf{l} = \epsilon) \\ \lor(\exists e.\ \lambda = \mathsf{B}\langle e\rangle \land \texttt{tid}(e) = \tau \land \mathsf{l} = \epsilon) \lor (\exists e.\ \lambda = \mathsf{PB}\langle e\rangle \land \mathsf{l} = \epsilon) \end{array}\right\}$$

Lemma A.7. *For all* $\mathsf{P}, \mathsf{S}, \mathsf{P}', \mathsf{S}'$*:*

- *for all* τ, l*, if* $\mathsf{P}, \mathsf{S} \xrightarrow{\tau:\mathsf{l}} \mathsf{P}', \mathsf{S}'$*, then there exists* λ *such that:* $((\tau, \mathsf{l}), \lambda) \in R_l$ *and* $\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}', \mathsf{S}'$*;*
- *for all* λ*, if* $\mathsf{P}, \mathsf{S} \xrightarrow{\lambda} \mathsf{P}', \mathsf{S}'$*, then there exists* τ, l *such that:* $((\tau, \mathsf{l}), \lambda) \in R_l$ *and* $\mathsf{P}, \mathsf{S} \xrightarrow{\tau:\mathsf{l}} \mathsf{P}', \mathsf{S}'$*.*

Proof. By straightforward induction on the structures of $\xrightarrow{\tau:\mathsf{l}}$ and $\xrightarrow{\lambda}$. □

Let

$$R_m \triangleq \left\{ \begin{array}{l|l} ((\mathsf{M}, \mathsf{PB}, \mathsf{B}), & (\mathsf{M}, \mathsf{PB}, \mathsf{B}) \in \textsc{Mem} \times \textsc{PBuff} \times \textsc{BMap} \\ \quad (M, PB, B)) & \wedge (M, PB, B) \in \textsc{AMem} \times \textsc{APBuff} \times \textsc{ABMap} \\ & \wedge \forall \mathtt{x}, v.\ \mathsf{M}(\mathtt{x}) = v \iff \mathtt{val_w}(M(\mathtt{x})) = v \\ & \wedge \mathsf{sim_{pb}}(\mathsf{PB}, PB) \wedge \mathsf{sim_b}(\mathsf{B}, B) \end{array} \right\}$$

$$\begin{aligned} \mathsf{sim_{pb}}(\mathsf{PB}, PB) \triangleq\ & \mathsf{PB} = PB = \epsilon \vee \\ & \exists pb, \mathsf{pb}, \mathsf{PB}', PB'.\ PB = PB'.(-, pb) \wedge \mathsf{PB} = \mathsf{PB}'.\mathsf{pb} \wedge \mathsf{sim_{pb}}(\mathsf{PB}', PB') \\ & \wedge \forall \mathtt{x}.\ \mathsf{sim_w}(\mathsf{pb}(\mathtt{x}), pb(\mathtt{x})) \end{aligned}$$

$$\mathsf{sim_w}(s_1, s_2) \triangleq (s_1 = s_2 = \epsilon) \vee (\exists v, s_1', s_2', e.\ s_1 = s_1'.v \wedge s_2 = s_2'.e \wedge \mathtt{val_w}(e) = v)$$

$$\mathsf{sim_b}(\mathsf{B}, B) \triangleq (\mathsf{B} = B = \epsilon) \vee (\exists \mathtt{x}, v, \mathsf{B}', B', e.\ \mathsf{B} = \mathsf{B}'.(\mathtt{x}, v) \wedge B = B'.e \wedge \mathtt{val_w}(e) = v \wedge \mathtt{loc}(e) = \mathtt{x})$$

Lemma A.8. *For all* $\mathsf{M}, \mathsf{PB}, \mathsf{B}, M, PB, B, M', PB', B'$*:*

- $((\mathsf{M}_0, \mathsf{PB}_0, \mathsf{B}_0), (M_0, PB_0, B_0)) \in R_m$*;*
- *for all* $\mathsf{M}', \mathsf{PB}', \mathsf{B}', \tau, \mathsf{l}$*, if* $((\mathsf{M}, \mathsf{PB}, \mathsf{B}), (M, PB, B)) \in R_m$ *and* $(\mathsf{M}, \mathsf{PB}, \mathsf{B}) \xrightarrow{\tau:\mathsf{l}} (\mathsf{M}', \mathsf{PB}', \mathsf{B}')$*, then there exist* M', PB', B', λ *such that* $((\tau, \mathsf{l}), \lambda) \in R_l$*,* $((\mathsf{M}', \mathsf{PB}', \mathsf{B}'), (M', PB', B')) \in R_m$ *and* $(M, PB, B) \xrightarrow{\lambda} (M', PB', B')$*.*
- *for all* M', PB', B', λ*, if* $((\mathsf{M}, \mathsf{PB}, \mathsf{B}), (M, PB, B)) \in R_m$ *and* $(M, PB, B) \xrightarrow{\lambda} (M', PB', B')$*, then there exist* $\mathsf{M}', \mathsf{PB}', \mathsf{B}', \tau, \mathsf{l}$ *such that* $((\tau, \mathsf{l}), \lambda) \in R_l$*,* $((\mathsf{M}', \mathsf{PB}', \mathsf{B}'), (M', PB', B')) \in R_m$ *and* $(\mathsf{M}, \mathsf{PB}, \mathsf{B}) \xrightarrow{\tau:\mathsf{l}} (\mathsf{M}', \mathsf{PB}', \mathsf{B}')$*.*

Proof. The proof of the first part follows immediately from the definitions of M_0, PB_0, B_0, M_0, PB_0, B_0. The proofs of the last two parts follow from straightforward induction on the structures of $\xrightarrow{\tau:\mathsf{l}}$ and $\xrightarrow{\lambda}$. □

Let

$$R \triangleq \left\{ \begin{array}{l|l} ((\mathsf{P}, \mathsf{S}, \mathsf{M}, \mathsf{PB}, \mathsf{B}), & \mathsf{P} \in \textsc{Prog} \wedge \mathsf{S} \in \textsc{SMap} \wedge \mathcal{H} \in \textsc{Hist} \wedge \pi \in \textsc{Path} \\ (\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi)) & \wedge ((\mathsf{M}, \mathsf{PB}, \mathsf{B}), (M, PB, B)) \in R_m \end{array} \right\}$$

Lemma A.9. *For all* $\mathsf{P}, \mathsf{M}, \mathsf{PB}, \mathsf{B}, M, PB, B, M', PB', B', \mathcal{H}, \pi$*:*

- $((\mathsf{P}, \mathsf{S}_0, \mathsf{M}_0, \mathsf{PB}_0, \mathsf{B}_0), (\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon)) \in R$*;*
- *for all* $\mathsf{P}', \mathsf{S}', \mathsf{M}', \mathsf{PB}', \mathsf{B}'$*, if* $((\mathsf{P}, \mathsf{S}, \mathsf{M}, \mathsf{PB}, \mathsf{B}), (\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi)) \in R$ *and* $(\mathsf{P}, \mathsf{S}, \mathsf{M}, \mathsf{PB}, \mathsf{B}) \Rightarrow (\mathsf{P}', \mathsf{S}', \mathsf{M}', \mathsf{PB}', \mathsf{B}')$*, then there exist* $M', PB', B', \mathcal{H}', \pi'$ *such that* $((\mathsf{P}', \mathsf{S}', \mathsf{M}', \mathsf{PB}', \mathsf{B}'), (\mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}', \pi')) \in R$ *and* $(\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi) \Rightarrow (\mathsf{P}, \mathsf{S}', M', PB', B', \mathcal{H}', \pi')$*.*
- *for all* $\mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}', \pi'$*, if* $((\mathsf{P}, \mathsf{S}, \mathsf{M}, \mathsf{PB}, \mathsf{B}), (\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi)) \in R$ *and* $(\mathsf{P}, \mathsf{S}, M, PB, B, \mathcal{H}, \pi) \Rightarrow (\mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}', \pi')$*, then there exist* $\mathsf{M}', \mathsf{PB}', \mathsf{B}'$ *such that* $((\mathsf{P}', \mathsf{S}', \mathsf{M}', \mathsf{PB}', \mathsf{B}')$ $, (\mathsf{P}', \mathsf{S}', M', PB', B', \mathcal{H}', \pi')) \in R$ *and* $(\mathsf{P}, \mathsf{S}, M, PB, B) \Rightarrow (\mathsf{P}', \mathsf{S}', M', PB', B')$*.*

Proof. The proof of the first part follows immediately from the definitions of R and Lemma A.8. The proofs of the last two parts follow from straightforward induction on the structures of $\xrightarrow{\tau:\mathsf{l}}$, $\xrightarrow{\lambda}$, Lemma A.7 and Lemma A.8. □

Theorem 6 (Intermediate and operational semantics equivalence)**.** *For all* P, S*:*

- *for all* M*, if* $\mathsf{P}, \mathsf{S}_0, \mathsf{M}_0, \mathsf{PB}_0, \mathsf{B}_0 \Rightarrow^* \textbf{skip}||\cdots||\textbf{skip}, \mathsf{S}, \mathsf{M}, \mathsf{PB}_0, \mathsf{B}_0$*, then there exist* $M, \mathcal{H}, \pi$ *such that* $\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \textbf{skip}||\cdots||\textbf{skip}, \mathsf{S}, M, PB_0, B_0, \mathcal{H}, \pi$ *and* $((\mathsf{M}, \mathsf{PB}_0, \mathsf{B}_0), (M, PB_0, B_0)) \in R_m$*;*
- *for all* $M, \mathcal{H}, \pi$*, if* $\mathsf{P}, \mathsf{S}_0, M_0, PB_0, B_0, \epsilon, \epsilon \Rightarrow^* \textbf{skip}||\cdots||\textbf{skip}, \mathsf{S}, M, PB_0, B_0, \mathcal{H}, \pi$*, then there exists* M *such that* $\mathsf{P}, \mathsf{S}_0, \mathsf{M}_0, \mathsf{PB}_0, \mathsf{B}_0 \Rightarrow^* \textbf{skip}||\cdots||\textbf{skip}, \mathsf{S}, \mathsf{M}, \mathsf{PB}_0, \mathsf{B}_0$ *and* $((\mathsf{M}, \mathsf{PB}_0, \mathsf{B}_0), (M, PB_0, B_0)) \in R_m$*.*

Proof. Follows from Lemma A.9 and straightforward induction on the length of $\Rightarrow^*$. □

B SIMPLE QUEUE LIBRARY

For an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$ of a program P with $G_i = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$, observe that when P comprises k threads, the trace of each execution era (via `start()` or `recover()`) comprises two stages: i) the trace of the *setup* stage by the master thread τ_0 performing initialisation or recovery, prior to the call to `run(P)`; followed (in po order) by ii) the trace of each of the constituent program threads $\tau_1 \cdots \tau_k$, provided that the execution did not crash during the setup stage.

Note that as the execution is PTSO-valid, thanks to the placement of the persistent fence operations (**pfence**), for each thread τ_j, we know that the set of persistent events in execution era i, namely E_i^P, contains roughly a *prefix* (in po order) of thread τ_j's trace. More concretely, for each constituent thread $\tau_j \in \{\tau_1 \cdots \tau_k\} = dom(\mathsf{P})$, there exist $P_1^j \cdots P_n^j$ such that:

1) $\mathsf{P}[\tau_j] = o_j^0; \cdots; o_j^{P_1^j}; o_j^{P_1^j+1}; \cdots o_j^{P_2^j}; \cdots; o_j^{P_{n-1}^j+1}; \cdots; o_j^{P_n^j}$, comprising enq and deq operations; and

2) at the beginning of each execution era $i \in \{1 \cdots n\}$, the program executed by thread τ_j (calculated in P' and subsequently executed by calling `run(P')`) is that of $\mathtt{sub}(\mathsf{P}[\tau_j], P_j^{i-1}{+}1)$, where $P_j^0 = -1$, for all j; and

3) in each execution era $i \in \{1 \cdots n\}$, the trace $H_{(i,j)}$ of each constituent thread $\tau_j \in dom(\mathsf{P})$ is of the following form:

$$
\begin{aligned}
H_{(i,j)} \triangleq\; & H(o_j^{P_j^{i-1}+1}, P_j^{i-1}{+}1, \tau_j, n_j^{P_j^{i-1}+1}) \xrightarrow{\mathsf{po}} \cdots \xrightarrow{\mathsf{po}} H(o_j^{P_j^i}, P_j^i, \tau_j, n_j^{P_j^i}) \\
& \xrightarrow{\mathsf{po}} H(o_j^{P_j^i+1}, P_j^i{+}1, \tau_j, n_j^{P_j^i+1}) \xrightarrow{\mathsf{po}} \cdots \xrightarrow{\mathsf{po}} H(o_j^{m_j^i-1}, m_j^i{-}1, \tau_j, n_j^{m_j^i-1}) \\
& \xrightarrow{\mathsf{po}} H'(o_j^{m_j^i}, m_j^i, \tau_j, n_j^{m_j^i})
\end{aligned}
$$

for some $m_j^i, n_j^{P_j^{i-1}+1}, \cdots, n_j^{P_j^i}, n_j^{P_j^i+1}, \cdots, n_j^{m_j^i}$, where:

- The first line denotes the execution of the $(P_j^{i-1}{+}1)^{\text{st}}$ to $(P_j^i)^{\text{th}}$ library calls of thread τ_j, with $H(o, \tau, p, n)$ defined shortly. Moreover, before crashing and proceeding to the next era, *all* volatile events (those in *PE*) in $H(o_j^{P_j^{i-1}+1}, P_j^{i-1}{+}1, \tau_j, n_j^{P_j^{i-1}+1}) \xrightarrow{\mathsf{po}} \cdots \xrightarrow{\mathsf{po}} H(o_j^{P_j^i-1}, P_j^i{-}1, \tau_j, n_j^{P_j^i-1})$ have persisted, and a *prefix* (in po order) of the volatile events in $H(o_j^{P_j^i}, P_j^i, \tau_j, n_j^{P_j^i})$ have persisted. Note that this prefix may be equal to $H(o_j^{P_j^i}, P_j^i, \tau_j, n_j^{P_j^i})$, in which case all its events have persisted.
- The second line denotes the execution of the subsequent library calls of thread τ_j where $m_j^i \leq P_j^n$, with *none* of their volatile events having persisted.
- The last line denotes the execution of the $(m_j^i)^{\text{th}}$ call of thread τ_j ($m_j^i \leq P_j^n$), during which the program crashed and thus the execution of era i ended. The $H'(o, \tau, p, n)$ denotes a (potentially full) prefix of $H(o, \tau, p, n)$.

The trace $H(o, \tau, p, n)$ of each library call is defined as follows:

$$
\begin{aligned}
H(\mathtt{deq()}, \tau, p, n) \triangleq\; & inv{=}\mathtt{I}(\iota_p, \mathtt{deq}, ()) \xrightarrow{\mathsf{po}} \mathtt{R}(pc, p) \xrightarrow{\mathsf{po}} \mathtt{R}(\mathtt{tid}_\tau, \tau) \\
& \xrightarrow{\mathsf{po}} \mathtt{R}(\mathtt{q.lock}, 1)^* \xrightarrow{\mathsf{po}} ql{=}\mathtt{U}(\mathtt{q.lock}, 0, 1) \\
& \xrightarrow{\mathsf{po}} r{=}\mathtt{R}(\mathtt{q.head}, h) \xrightarrow{\mathsf{po}} r{=}\mathtt{R}(\mathtt{q.data[h]}, n) \\
& \xrightarrow{\mathsf{po}} lin_1{=}\mathtt{W}(\mathtt{map[\tau]}, (p, n)) \xrightarrow{\mathsf{po}} S_1 \xrightarrow{\mathsf{po}} \mathsf{PF} \xrightarrow{\mathsf{po}} S_2 \\
& \xrightarrow{\mathsf{po}} qu{=}\mathtt{W}(\mathtt{q.lock}, 0) \xrightarrow{\mathsf{po}} ack{=}\mathtt{A}(\iota_p, \mathtt{deq}, n)
\end{aligned}
$$

with

$$S_1 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ \mathtt{R}(n.\mathtt{t}, \tau') \xrightarrow{\text{po}} \mathtt{R}(n.\mathtt{pc}, p') \xrightarrow{\text{po}} \mathtt{R}(\mathtt{map}[\tau'], (tp, tn)) \xrightarrow{\text{po}} S_3 & \text{otherwise} \end{cases}$$

$$S_3 = \begin{cases} \emptyset & \text{if } tp > p' \\ \mathtt{U}(\mathtt{map}[\tau'], (tp', tn'), (p'{+}1, \bot)) & \text{if } tp \leq p' \text{ and } (tp', tn'){=}(tp, tn) \\ \mathtt{R}(\mathtt{map}[\tau'], (tp', tn')) & \text{otherwise} \end{cases}$$

$$S_2 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ lin_2{=}\mathtt{W}(\mathtt{q.head}, h{+}1) \xrightarrow{\text{po}} \mathsf{PF} & \text{otherwise} \end{cases}$$

for some $\tau', p', tp, tn, tp', tn'$; and

$$\begin{aligned} H(\mathtt{enq}(v), \tau, p, n) \triangleq\ & inv{=}\mathtt{I}(\iota_p, \mathtt{enq}, n) \xrightarrow{\text{po}} \mathtt{R}(pc, p) \xrightarrow{\text{po}} \mathtt{R}(\mathtt{tid}_\tau, \tau) \\ & \xrightarrow{\text{po}} \mathtt{W}(n\mathtt{.val}, v) \xrightarrow{\text{po}} \mathtt{W}(n\mathtt{.tid}, \tau) \xrightarrow{\text{po}} \mathtt{W}(n\mathtt{.pc}, p) \\ & \xrightarrow{\text{po}} \mathtt{W}(\mathtt{map}[\tau], (p, n)) \xrightarrow{\text{po}} \mathsf{PF} \\ & \xrightarrow{\text{po}} \mathtt{R}(\mathtt{q.lock}, 1)^* \xrightarrow{\text{po}} \mathtt{U}(\mathtt{q.lock}, 0, 1) \xrightarrow{\text{po}} \mathtt{R}(\mathtt{q.head}, h) \\ & \underbrace{\xrightarrow{\text{po}} \mathtt{R}(\mathtt{q.data}[h], v_0) \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} \mathtt{R}(\mathtt{q.data}[h{+}s{-}1], v_{s-1})}_{s \text{ times}} \\ & \xrightarrow{\text{po}} \mathtt{R}(\mathtt{q.data}[h{+}s], \mathtt{null}) \xrightarrow{\text{po}} lin{=}\mathtt{W}(\mathtt{q.data}[h{+}s], n) \\ & \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} \mathtt{W}(\mathtt{q.lock}, 0) \xrightarrow{\text{po}} ack{=}\mathtt{A}(\iota_p, \mathtt{enq}, ()) \end{aligned}$$

for some $s \geq 0$, and for all $v \in \{v_0 \cdots v_{s-1}\}$, $v \neq \mathtt{null}$. In the above traces, for brevity we have omitted the thread identifiers (τ_j) and event identifiers and represent each event with its label only. We use the $H(\mathtt{enq}(\text{-}), \tau, p, n)$ prefix to extract its specific events, e.g. $H(\mathtt{enq}(\text{-}), p, n).inv$.

It is straightforward to demonstrate that $\mathsf{hb}_i = (\mathsf{po}_i \cup \mathsf{rf}_i)^+$ restricted to the lock events in $\bigcup_{\tau_j \in dom(\mathsf{P})} \bigcup_{l=P_j^{i-1}+1}^{m_j^i} \{H(o_j^l, \tau_j, l, n_j^l).ql, H(o_j^l, \tau_j, l, n_j^l).qu\}$ is a strict total order.

In particular, we know there exists an enumeration $C_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1). \cdots . H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i})$ of $\bigcup_{\tau_j \in dom(\mathsf{P})} \{H(o_j^{P_j^{i-1}+1}, \tau_j, P_j^{i-1}{+}1, n_j^{P_j^{i-1}+1}) \cdots H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}\}$, such that:

$$\left\{ \begin{matrix} (H(c_i^k, \tau_i^k, p_i^k, n_i^k).ql, H(c_i^k, \tau_i^k, p_i^k, n_i^k).qu), \\ (H(c_i^l, \tau_i^l, p_i^l, n_i^l).qu, H(c_i^{l+1}, \tau_i^{l+1}, p_i^{l+1}, n_i^{l+1}).ql) \end{matrix} \middle| \begin{matrix} k \in \{1 \cdots t_i\} \\ \wedge\ l \in \{1 \cdots t_i{-}1\} \end{matrix} \right\}^+ \subseteq (\mathsf{hb}_i)_{loc}|_{\text{imm}}$$

$$\text{Let } \mathtt{lp}(H(o, \tau, p, n)) \triangleq \begin{cases} H(o, \tau, p, n).lin & \text{if } o{=}\mathtt{enq}(v) \\ H(o, \tau, p, n).lin_1 & \text{if } o{=}\mathtt{deq}() \text{ and } H(o, \tau, p, n).S_2{=}\emptyset \\ H(o, \tau, p, n).lin_2 & \text{if } o{=}\mathtt{deq}() \text{ and } H(o, \tau, p, n).S_2{\neq}\emptyset \end{cases}$$

For each $\tau_j \in dom(\mathsf{P})$ let:

$$E_{(i,j)}^P = E_i^P \cap \left\{ e \,\middle|\, \mathtt{tid}(e) = \tau_j \right\} \qquad E'_{(i,j)} = E_{(i,j)}^P \cup S_{(i,j)}$$

where

$$S_{(i,j)} \triangleq \left\{ \mathsf{A}(\iota, m, r) \;\middle|\; \begin{array}{l} \exists o, a, p, n, inv. \\ \quad inv = \mathtt{I}(\iota, m, a) = \max\left(\textsf{nvo}|_{E^P_{(i,j)} \cap I}\right) \\ \quad \land\, inv \in H(o, \tau_j, p, n) \land \forall r'.\, \mathsf{A}(\iota, m, r') \notin H(o, \tau_j, p, n) \\ \quad \land\, \mathtt{lp}(H(o, \tau_j, p, n)) \in E^P_{(i,j)} \\ \quad \land\, (m{=}\mathtt{deq} \Rightarrow r{=}n) \land (m{=}\mathtt{enq} \Rightarrow r{=}()) \end{array} \right\}$$

Let $E'_i = \bigcup_{\tau_j \in dom(\mathrm{P})} E'_{(i,j)}$. From the definition of each $E'_{(i,j)}$ and $E^P_{(i,j)}$ we then know that $E^P_i \subseteq E'_i$ and $E'_i \in \mathtt{comp}(E^P_i)$. Let $T_i = \mathtt{trunc}(E'_i)$ and

$$\begin{array}{l} H_i = H(c^1_i, \tau^1_i, p^1_i, n^1_i).inv \,.\, H(c^1_i, \tau^1_i, p^1_i, n^1_i).ack \\ \quad \cdots .H(c^{t_i}_i, \tau^{t_i}_i, p^{t_i}_i, n^{t_i}_i).inv \,.\, H(c^{t_i}_i, \tau^{t_i}_i, p^{t_i}_i, n^{t_i}_i).ack \end{array}$$

Let

$$\begin{array}{rl} \mathsf{isQ}(q, Q, \textsf{nvo}, E^0, E^P) \triangleq & (init_q = \max\left(\textsf{nvo}|_{E^P \cap (W \cup U)_q}\right) \land Q{=}\epsilon) \\ & \lor(\exists h, s.\; |Q| = s \land \forall v \in Q.\; v \neq \mathtt{null} \\ & \quad \land \mathsf{val_w}(\max\left(\textsf{nvo}|_{E^P \cap (W \cup U)_{q.\mathtt{head}}}\right)){=}h \\ & \quad \land \forall k \in \{0 \cdots s{-}1\}. \\ & \qquad \mathsf{val_w}(\max\left(\textsf{nvo}|_{E^P \cap (W \cup U)_{q.\mathtt{data}[h+k]}}\right)) {=}\, Q|_k \\ & \quad \land \forall k \geq s. \\ & \qquad \mathsf{val_w}(\max\left(\textsf{nvo}|_{E^0 \cap (W \cup U)_{q.\mathtt{data}[h+k]}}\right)){=}\mathtt{null} \\ & \qquad \land (E^P \setminus E^0) \cap (W \cup U)_{q.\mathtt{data}[h+k]} = \emptyset) \end{array}$$

and

$$\mathtt{getQ}(s, H) \triangleq \begin{cases} s & \text{if } H{=}\epsilon \\ \mathtt{getQ}(s; n, H') & \text{if } \exists n, H', \iota.\; n{\neq}\mathtt{null} \land H{=}\mathtt{I}(\iota, \mathtt{enq}, n).\mathsf{A}(\iota, \mathtt{enq}, ()).H' \\ \mathtt{getQ}(s', H') & \text{if } \exists n, H', \iota, s'.\; n{\neq}\mathtt{null} \land s{=}n; s' \\ & \qquad\qquad \land H{=}\mathtt{I}(\iota, \mathtt{deq}, ()).\mathsf{A}(\iota, \mathtt{deq}, n).H' \\ \mathtt{getQ}(s, H') & \text{if } \exists H', \iota.\; s{=}\epsilon \land H{=}\mathtt{I}(\iota, \mathtt{deq}, ()).\mathsf{A}(\iota, \mathtt{deq}, \mathtt{null}).H' \\ \text{undefined} & \text{otherwise} \end{cases}$$

Lemma B.1. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above with $C_i = H(c^1_i, \tau^1_i, p^1_i, n^1_i). \cdots .H(c^{t_i}_i, \tau^{t_i}_i, p^{t_i}_i, n^{t_i}_i)$. For all $i \in \{1 \cdots n\}$, and a, b, let $O^b_a = H(c^a_i, \tau^a_i, p^a_i, n^a_i).inv.H(c^a_i, \tau^a_i, p^a_i, n^a_i).ack. \cdots .H(c^b_i, \tau^b_i, p^b_i, n^b_i).inv.H(c^b_i, \tau^b_i, p^b_i, n^b_i).ack$.*

For all $G_i = (E^0_i, E^P_i, E_i, \mathsf{po}_i, \textsf{rf}_i, \textsf{tso}_i, \textsf{nvo}_i)$, H_i, for all Q^0_i and for all $l \in \{0 \cdots t_i\}$, $k{=}t_i{-}l$, $E^k_i = E^P_i \setminus \bigcup_{x=k+1}^{t_i} H(c^x_i, \tau^x_i, p^x_i, n^x_i).E$, and Q^k_i:

$$\begin{array}{c} \mathtt{getQ}(Q^0_i, O^k_1) = Q^k_i \land \mathsf{isQ}(\mathsf{q}, Q^k_i, \textsf{nvo}_i, E^0_i, E^k_i) \Rightarrow \\ \exists Q^t_i.\; \mathtt{getQ}(Q^k_i, O^{t_i}_{k+1}) = Q^t_i \land \mathsf{isQ}(\mathsf{q}, Q^t_i, \textsf{nvo}_i, E^0_i, E^P_i) \end{array}$$

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$. Let H_i and C_i be as defined as above for all $i \in \{1 \cdots n\}$. Pick an arbitrary $i \in \{1 \cdots n\}$, $G_i = (E^0_i, E^P_i, E_i, \mathsf{po}_i, \textsf{rf}_i, \textsf{tso}_i, \textsf{nvo}_i)$ and H_i. We proceed by induction on l.

Base case $l = 0, k = t_i$
Pick arbitrary Q_i^0 and Q_i^k such that $\texttt{getQ}(Q_i^0, O_1^k) = Q_i^k$ and $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^k)$. As $k = t_i$, we have $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^P)$. As $O_{k+1}^{t_i}=\epsilon$, we have $\texttt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^k$, as required.

Inductive case $0 < l \leq t_i$

$$\begin{array}{l}\forall Q.\ \forall k' > k.\ \texttt{getQ}(Q_i^0, O_1^{k'}) = Q \wedge \text{isQ}(\text{q}, Q, \text{nvo}_i, E_i^0, E_i^{k'}) \Rightarrow \\ \quad \exists Q_i^t.\ \texttt{getQ}(Q, O_{k'+1}^{t_i}) = Q_i^t \wedge \text{isQ}(\text{q}, Q_i^t, \text{nvo}_i, E_i^0, E_i^P)\end{array} \tag{I.H.}$$

Pick arbitrary Q_i^0 and Q_i^k such that $\texttt{getQ}(Q_i^0, O_1^k) = Q_i^k$ and $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^k)$. We are then required to show that there exists Q_i^t such that $\texttt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t$ and $\text{isQ}(\text{q}, Q_i^t, \text{nvo}_i, E_i^0, E_i^P)$. We then know:

$$O_{k+1}^{t_i} = H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1}).inv.H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1}).ack.O_{k+2}^{t_i}$$

There are now three cases to consider: 1) there exists m such that c_i^{k+1}=$\texttt{enq}(m)$ and n_i^{k+1}=m; or 2) there exists $m \neq \texttt{null}$ such that c_i^{k+1}=$\texttt{deq()}$ and n_i^{k+1}=m; or 3) c_i^{k+1}=$\texttt{deq()}$ and n_i^{k+1}=$\texttt{null}$.

In case (1), as $\texttt{getQ}(Q_i^0, O_1^k) = Q_i^k$, from its definition we have $\texttt{getQ}(Q_i^0, O_1^{k+1}) = Q_i^k.m$. Let $Q_i^{k+1} = Q_i^k.m$. Given $H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1})$, since from the PTSO-validity of G_i we have $E_i^0 \times (E_i^P \setminus E_i^0) \subseteq \text{nvo}_i$ and as $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^k)$ holds, from its definition we have $\text{isQ}(\text{q}, Q_i^{k+1}, \text{nvo}_i, E_i^0, E_i^{k+1})$. From (I.H.) we know there exists Q_i^t such that $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$ and $\text{isQ}(\text{q}, Q_i^t, \text{nvo}_i, E_i^0, E_i^P)$. As $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$, from its definition we also have $\texttt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t$, as required.

In case (2), given the trace of $H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1})$ we know that there exists w, r, a such that w=$\text{W}(\texttt{q.data}[a], m)$, $r = H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1}).r$ and $(w, r) \in \text{rf}_i$. As hb_i is acyclic and G_i is PTSO-valid, we know either:

i) $w \in E_i^0$ and for all $j \in \{1 \cdots k\}$, $H(c_i^j, \tau_i^j, p_i^j, n_i^j).E \cap (W \cup U)_{\texttt{q.data}[a]} = \emptyset$; or

ii) exists j s.t. $1 \leq j \leq k$ and $w \in H(c_i^j, \tau_i^j, p_i^j, n_i^j)$ and for all $j' \in \{j{+}1 \cdots k\}$, $H(c_i^{j'}, \tau_i^{j'}, p_i^{j'}, n_i^{j'}).E \cap (W \cup U)_{\texttt{q.data}[a]} = \emptyset$.

As $E_i^0 \subseteq E_i^P$ and the events of $H(c_i^j, \tau_i^j, p_i^j, n_i^j)$ are persistent (discussed above in the construction of H_i), we know that $w \in E_i^k$. Moreover, as the lock events are totally ordered by hb_i, and $\text{hb}_i \subseteq \text{po} \cup \text{tso}$ (Lemma E.2), given the placement of $\texttt{pfence}$ instructions and the construction of the enumeration C_i, we know that for all locations x, if $w_1 = \text{W}(\text{x}, -) \in H(c_i^f, -, -, -)$, $w_2 = \text{W}(\text{x}, -) \in H(c_i^g, -, -, -)$, and $f < g$, then $(w_1, w_2) \in \text{nvo}_i$. As such, in both cases we know that $\max\left(\text{nvo}|_{E_i^k \cap (W \cup U)_{\texttt{q.data}[a]}}\right) = w$. Moreover, since $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^k)$ holds, we know that $\text{val}_{\text{w}}\left(\max\left(\text{nvo}|_{E_i^k \cap W_{\texttt{q.data}[a]}}\right)\right) = Q_i^k\big|_0$. We thus have $Q_i^k\big|_0 = m$.

Let $Q_i^k = m.Q'$ for some Q' and let $Q_i^{k+1} = Q'$. As $\texttt{getQ}(Q_i^0, O_1^k)$ holds, from its definition we also have $\texttt{getQ}(Q_i^0, O_1^{k+1}) = Q_i^{k+1}$. Given the trace $H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1})$, as $\text{isQ}(\text{q}, Q_i^k, \text{nvo}_i, E_i^0, E_i^k)$ holds, from its definition we have $\text{isQ}(\text{q}, Q_i^{k+1}, \text{nvo}_i, E_i^0, E_i^{k+1})$. From (I.H.) we then know there exists Q_i^t such that $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$ and $\text{isQ}(\text{q}, Q_i^t, \text{nvo}_i, E_i^0, E_i^P)$. As $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$, from its definition we also have $\texttt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t$, as required.

Case (3) is analogous to that of case (2) and is omitted here.

□

Corollary 2. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above. For all $G_i = (E_i^0, E_i^P, E_i, \text{po}_i, \text{rf}_i, \text{tso}_i, \text{nvo}_i)$, H_i and for all Q_i^0:*

$$\text{isQ}(\text{q}, Q_i^0, \text{nvo}_i, E_i^0, E_i^0) \Rightarrow$$

$$\exists Q_i^t.\ \mathtt{getQ}(Q_i^0, H_i) = Q_i^t \land \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)$$

Proof. Follows immediately from the previous lemma when $k = 0$. □

Lemma B.2. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, if $H = H_1. \cdots .H_n$ with H_i defined as above for all $i \in \{1 \cdots n\}$, then:*

$$\exists Q.\ \mathtt{getQ}(\epsilon, H) = Q$$

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, with $H = H_1. \cdots .H_n$ and H_i defined as above for all $i \in \{1 \cdots n\}$. Let $Q_1^0 = \epsilon$. By definition we then have $\mathsf{isQ}(\mathsf{q}, Q_1^0, \mathsf{nvo}_1, E_1^0, E_1^0)$. On the other hand from Corollary 2 we have:

$$\begin{array}{l}
\exists Q_1^t.\ \mathtt{getQ}(Q_1^0, H_1) = Q_1^t \land \mathsf{isQ}(\mathsf{q}, Q_1^t, \mathsf{nvo}_1, E_1^0, E_1^P) \\
\forall Q_2^0.\ \mathsf{isQ}(\mathsf{q}, Q_2^0, \mathsf{nvo}_2, E_2^0, E_2^0) \Rightarrow \\
\quad \exists Q_2^t.\ \mathtt{getQ}(Q_2^0, H_2) = Q_2^t \land \mathsf{isQ}(\mathsf{q}, Q_2^t, \mathsf{nvo}_2, E_2^0, E_2^P) \\
\cdots \\
\forall Q_n^0.\ \mathsf{isQ}(\mathsf{q}, Q_n^0, \mathsf{nvo}_n, E_n^0, E_n^0) \Rightarrow \\
\quad \exists Q_n^t.\ \mathtt{getQ}(Q_n^0, H_n) = Q_n^t \land \mathsf{isQ}(\mathsf{q}, Q_n^t, \mathsf{nvo}_n, E_n^0, E_n^P)
\end{array}$$

For all $j \in \{2 \cdots n\}$, let $Q_j^0 = \mathtt{getQ}(Q_{j-1}^0, H_{j-1})$. From above we then have :

$$\begin{array}{l}
\exists Q_1^t, \cdots, Q_n^t. \\
\quad \mathtt{getQ}(Q_1^0, H_1) = Q_1^t \land \mathtt{getQ}(Q_1^t, H_2) = Q_2^t \land \cdots \land \mathtt{getQ}(Q_{n-1}^t, H_n) = Q_n^t
\end{array}$$

From its definition we thus know there exists Q_n^t such that $\mathtt{getQ}(Q_1^0, H_1. \cdots .H_n) = Q_n^t$. That is, there exists Q such that $\mathtt{getQ}(\epsilon, H) = Q$, as required. □

Theorem 7. *For all client programs* P *of the queue library (comprising calls to* `enq` *and* `deq` *only) and all PTSO-valid executions $\mathcal{E}$ of* `start(P)`*, $\mathcal{E}$ is persistently linearisable.*

Proof. Pick an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$ of P. For each $i \in \{1 \cdots n\}$, construct T_i and H_i as above. It then suffices to show that:

$$\forall i \in \{1 \cdots n\}.\ \forall a, b \in T_i.\ (a, b) \in \mathsf{hb}_i \Rightarrow a \prec_{H_i} b \tag{34}$$

$$\mathsf{fifo}(\epsilon, H) \text{ holds when } H = H_1. \cdots .H_n \tag{35}$$

TS. (34)
Pick arbitrary $i \in \{1 \cdots n\}$, $a, b \in T_i$ such that $(a, b) \in \mathsf{hb}_i$. We then know there exist $c, \tau, p, n, c', \tau', p', n'$ such that $a \in H(c, \tau, p, n)$, $b \in H(c', \tau', p', n')$ and either:
1) $H(c, \tau, p, n){=}H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).inv$ and $b = H(c, \tau, p, n).ack$; or
2) $H(c, \tau, p, n){=}H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).ack$ and $b = H(c, \tau, p, n).inv$; or
3) $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).inv$ and $b{=}H(c', \tau', p', n').ack$; or
4) $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).inv$ and $b{=}H(c', \tau', p', n').inv$; or
5) $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).ack$ and $b{=}H(c', \tau', p', n').inv$; or
6) $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, $a{=}H(c, \tau, p, n).ack$ and $b{=}H(c', \tau', p', n').ack$.

In case (1) the desired result holds immediately. In case (2) we have $b \xrightarrow{\mathsf{po}_i} a \xrightarrow{\mathsf{hb}_i} b$, and since $\mathsf{po}_i \subseteq \mathsf{hb}_i$ we have $b \xrightarrow{\mathsf{hb}_i} a \xrightarrow{\mathsf{hb}_i} b$. Consequently, from the transitivity of hb_i we have $(b, b) \in \mathsf{hb}_i$, contradicting the acyclicity of hb_i in Lemma E.1.

In case (3) from the totality of hb_i on lock events (see above), we know that either i) $(H(c, \tau, p, n).qu, H(c', \tau', p', n').ql) \in \mathsf{hb}_i$; or ii) $(H(c', \tau', p', n').qu, H(c, \tau, p, n).ql) \in \mathsf{hb}_i$. In case (3.i) from the construction of C_i we know that $a \prec_{H_i} b$, as required.

In case (3.ii), as $(a, b) \in \text{hb}_i$ and $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, we know there exists w, r, d, e, w', r' such that either:

a) $d \notin H(c, \tau, p, n)$, $e \notin H(c', \tau', p', n')$ and $a \xrightarrow{\text{po}_i} d \xrightarrow{\text{hb}_i} e \xrightarrow{\text{po}_i} b$; or

b) $w \in W \cap H(c, \tau, p, n)$, $e \notin H(c', \tau', p', n')$ and $a \xrightarrow{\text{po}_i} H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i{}^*} w \xrightarrow{\text{rf}_i} r \xrightarrow{\text{hb}_i} e \xrightarrow{\text{po}_i} b$; or

c) $r \in R \cap H(c', \tau', p', n')$, $d \notin H(c, \tau, p, n)$ and $a \xrightarrow{\text{po}_i} d \xrightarrow{\text{hb}_i{}^*} w \xrightarrow{\text{rf}_i} r \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu \xrightarrow{\text{po}_i} b$; or

d) $w \in W \cap H(c, \tau, p, n)$, $r \in R \cap H(c', \tau', p', n')$ and $a \xrightarrow{\text{po}_i} H(c, \tau, p, n).ql \xrightarrow{\text{po}_i} w \xrightarrow{\text{rf}_i} r' \xrightarrow{\text{hb}_i} w' \xrightarrow{\text{rf}_i} r \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu \xrightarrow{\text{po}_i} b$.

We next demonstrate that in all four cases (a-d) we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu$. We then have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu \xrightarrow{\text{hb}_i} H(c, \tau, p, n).ql$, and thus from the transitivity of hb_i we have $(H(c, \tau, p, n).ql, H(c, \tau, p, n).ql) \in \text{hb}_i$, contradicting the acyclicity of hb_i in Lemma E.1.

In case (3.ii.a) we also have $H(c, \tau, p, n).ql \xrightarrow{\text{po}_i} d$ and $e \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu$. As such we have $H(c, \tau, p, n).ql \xrightarrow{\text{po}_i} d \xrightarrow{\text{hb}_i} e \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu$, i.e. from the transitivity of hb_i we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu$. In case (3.ii.b) we also have $e \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu$. As such we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i{}^*} w \xrightarrow{\text{rf}_i} r \xrightarrow{\text{hb}_i} e \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu$, i.e. from the transitivity of hb_i we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu$. In case (3.ii.c) we also have $H(c, \tau, p, n).ql \xrightarrow{\text{po}_i} d$. As such we have $H(c, \tau, p, n).ql \xrightarrow{\text{po}_i} d \xrightarrow{\text{hb}_i{}^*} w \xrightarrow{\text{rf}_i} r \xrightarrow{\text{po}_i} H(c', \tau', p', n').qu$, i.e. from the transitivity of hb_i we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu$. In case (3.ii.d) from the transitivity of hb_i we have $H(c, \tau, p, n).ql \xrightarrow{\text{hb}_i} H(c', \tau', p', n').qu$.

In case (4) we then have $a \xrightarrow{\text{hb}_i} b \xrightarrow{\text{po}_i} H(c', \tau', p', n').ack$, and thus as $\text{po}_i \subseteq \text{hb}_i$ and hb_i is transitively closed, we have $a \xrightarrow{\text{hb}_i} H(c', \tau', p', n').ack$. As such, from the proof of part (3) we have $a \prec_{H_i} H(c', \tau', p', n').ack$, and consequently since $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, from the construction H_i we have $a \prec_{H_i} b$, as required.

In case (5) we then have $H(c, \tau, p, n).inv \xrightarrow{\text{po}_i} a \xrightarrow{\text{hb}_i} b \xrightarrow{\text{po}_i} H(c', \tau', p', n').ack$, and thus as $\text{po}_i \subseteq \text{hb}_i$ and hb_i is transitively closed, we have $H(c, \tau, p, n).inv \xrightarrow{\text{hb}_i} H(c', \tau', p', n').ack$. As such, from the proof of part (3) we have $H(c, \tau, p, n).inv \prec_{H_i} H(c', \tau', p', n').ack$, and consequently since $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, from the construction H_i we have $a \prec_{H_i} b$, as required.

In case (6) we then have $H(c, \tau, p, n).inv \xrightarrow{\text{po}_i} a \xrightarrow{\text{hb}_i} b$, and thus as $\text{po}_i \subseteq \text{hb}_i$ and hb_i is transitively closed, we have $H(c, \tau, p, n).inv \xrightarrow{\text{hb}_i} b$. As such, from the proof of part (3) we have $H(c, \tau, p, n).inv \prec_{H_i} b$, and consequently since $H(c, \tau, p, n) \neq H(c', \tau', p', n')$, from the construction H_i we have $a \prec_{H_i} b$, as required.

TS. (35)
From Lemma B.2 we know there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. From the definition of $\text{fifo}(.,.)$ we know $\text{fifo}(\epsilon, H)$ holds if and only if there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. As such we have $\text{fifo}(\epsilon, H)$, as required.

□

C MICHAEL-SCOTT QUEUE LIBRARY

As before, for an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$ of P with $G_i = (E^0, E^P, E, \text{po}, \text{rf}, \text{tso}, \text{nvo})$, observe that when P comprises k threads, the trace of each execution era (via `start()` or `recover()`) comprises two stages: i) the trace of the *setup* stage by the master thread τ_0 performing initialisation or recovery, prior to the call to `run(P)`; followed (in po order) by ii) the trace of each of the constituent program threads $\tau_1 \cdots \tau_k$, provided that the execution did not crash during the setup stage.

As before, thanks to the placement of the persistent fence operations (**pfence**), for each thread τ_j, we know that the set of persistent events in execution era i, namely E_i^P, contains roughly a *prefix* (in po order) of thread τ_j's trace. More concretely, for each constituent thread $\tau_j \in \{\tau_1 \cdots \tau_k\} = dom(\mathtt{P})$, there exist $P_j^1 \cdots P_j^n$ such that:

1) $\mathtt{P}[\tau_j] = o_j^0; \cdots ; o_j^{P_j^1}; o_j^{P_j^1+1}; \cdots o_j^{P_j^2}; \cdots ; o_j^{P_j^{n-1}+1}; \cdots ; o_j^{P_j^n}$, comprising `enq` and `deq` operations; and

2) at the beginning of each execution era $i \in \{1 \cdots n\}$, the program executed by thread τ_j (calculated in `P'` and subsequently executed by calling `run(P')`) is that of $\mathtt{sub(P[}\tau_j\mathtt{]},P_j^{i-1}\mathtt{+1)}$, where $P_j^0 = -1$, for all j; and

3) in each execution era $i \in \{1 \cdots n\}$, the trace $H_{(i,j)}$ of each constituent thread $\tau_j \in dom(\mathtt{P})$ is of the following form:

$$
\begin{aligned}
H_{(i,j)} \triangleq\ & H(o_j^{P_j^{i-1}+1}, \tau_j, P_j^{i-1}{+}1, n_j^{P_j^{i-1}+1}, e_j^{P_j^{i-1}+1}) \\
& \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i}) \\
& \xrightarrow{\text{po}} H(o_j^{P_j^i+1}, \tau_j, P_j^i{+}1, n_j^{P_j^i+1}, e_j^{P_j^i+1}) \\
& \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{m_j^i-1}, \tau_j, m_j^i{-}1, n_j^{m_j^i-1}, e_j^{m_j^i-1}) \\
& \xrightarrow{\text{po}} H'(o_j^{m_j^i}, \tau_j, m_j^i, n_j^{m_j^i}, e_j^{m_j^i})
\end{aligned}
$$

for some $m_j^i, n_j^{P_j^{i-1}+1}, \cdots, n_j^{P_j^i}, n_j^{P_j^i+1}, \cdots, n_j^{m_j^i}, e_j^{P_j^{i-1}+1}, \cdots, e_j^{P_j^i}, e_j^{P_j^i+1}, \cdots, e_j^{m_j^i}$ where:

- The first two lines denote the execution of the $(P_j^{i-1}{+}1)^{\text{st}}$ to $(P_j^i)^{\text{th}}$ library calls of thread τ_j, with $H(o, \tau, p, n, e)$ defined shortly. Moreover, before crashing and proceeding to the next era, *all* volatile events (those in *PE*) in $H(o_j^{P_j^{i-1}+1}, \cdots) \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{P_j^i-1}, \cdots)$ have persisted, and a *prefix* (in po order) of the volatile events in $H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i})$ have persisted. Note that this prefix may be equal to $H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i})$, in which case all its events have persisted.
- The next two lines denote the execution of the subsequent library calls of thread τ_j where $m_j^i \leq P_j^n$, with *none* of their volatile events having persisted.
- The last line denotes the execution of the $(m_j^i)^{\text{th}}$ call of thread τ_j ($m_j^i \leq P_j^n$), during which the program crashed and thus the execution of era i ended. As before, the $H'(o, \tau, p, n, e)$ denotes a (potentially full) prefix of $H(o, \tau, p, n, e)$.

The trace $H(o, \tau, p, n, e)$ of each library call is defined as follows:

$$
\begin{aligned}
H(\mathtt{deq}(), \tau, p, n, h) \triangleq\ & inv{=}\mathtt{I}(\iota_p, \mathtt{deq}, ()) \xrightarrow{\text{po}} \mathtt{R}(pc, p) \xrightarrow{\text{po}} \mathtt{R}(\mathtt{tid}_\tau, \tau) \xrightarrow{\text{po}} FE \\
& \xrightarrow{\text{po}} r_h{=}\mathtt{R}(\mathtt{q.head}, h) \xrightarrow{\text{po}} r{=}\mathtt{R}(\mathtt{q.data[}h\mathtt{]}, n) \\
& \xrightarrow{\text{po}} S_0 \xrightarrow{\text{po}} lin_1{=}\mathtt{W}(\mathtt{map[}\tau\mathtt{][}p\mathtt{]}, n) \xrightarrow{\text{po}} S_1 \xrightarrow{\text{po}} \mathtt{PF} \xrightarrow{\text{po}} S_2 \\
& \xrightarrow{\text{po}} ack{=}\mathtt{A}(\iota_p, \mathtt{deq}, n)
\end{aligned}
$$

where *FE* denotes the sequence of events, attempting but failing to set the `rem` field of the head node, with

$$
S_0 = \begin{cases} \emptyset & \text{if } n = \mathtt{null} \\ \mathtt{U}(n.\mathtt{rem}, \mathtt{null}, \tau) & \text{otherwise} \end{cases}
$$

$$S_1 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ \mathsf{R}(n.\texttt{t}, \tau') \xrightarrow{\text{po}} \mathsf{R}(n.\texttt{pc}, p') \xrightarrow{\text{po}} \mathsf{W}(\texttt{map}[\tau'][p'], \top) & \text{otherwise} \end{cases}$$

$$S_2 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ lin_2{=}\mathsf{W}(\texttt{q.head}, h{+}1) \xrightarrow{\text{po}} \mathsf{PF} & \text{otherwise} \end{cases}$$

for some τ', p'; and

$$\begin{aligned} H(\texttt{enq}(v), \tau, p, n, e) \triangleq\ & inv{=}\mathsf{I}(\iota_p, \texttt{enq}, n) \xrightarrow{\text{po}} \mathsf{R}(pc, p) \xrightarrow{\text{po}} \mathsf{R}(\texttt{tid}_\tau, \tau) \\ & \xrightarrow{\text{po}} \mathsf{W}(n.\texttt{val}, v) \xrightarrow{\text{po}} \mathsf{W}(n.\texttt{tid}, \tau) \xrightarrow{\text{po}} \mathsf{W}(n.\texttt{pc}, p) \xrightarrow{\text{po}} \mathsf{W}(n.\texttt{rem}, \texttt{null}) \\ & \xrightarrow{\text{po}} \mathsf{W}(\texttt{map}[\tau][p], n) \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} \mathsf{R}(\texttt{q.head}, h) \\ & \underbrace{\xrightarrow{\text{po}} \mathsf{R}(\texttt{q.data}[h], v_0) \xrightarrow{\text{po}} A_0 \cdots \mathsf{R}(\texttt{q.data}[h{+}s{-}1], v_{s-1}) \xrightarrow{\text{po}} A_{s-1}}_{s \text{ times}} \\ & \xrightarrow{\text{po}} \mathsf{R}(\texttt{q.data}[h{+}s], \texttt{null}) \xrightarrow{\text{po}} lin{=}\mathsf{U}(\texttt{q.data}[h{+}s], \texttt{null}, n) \\ & \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} ack{=}\mathsf{A}(\iota_p, \texttt{enq}, ()) \end{aligned}$$

for some $s \geq 0$ such that $h{+}s = e$, and for all $k \in \{0 \cdots s{-}1\}$, either 1) $v_k \neq \texttt{null}$ and $A_k = \emptyset$; or $v_k = \texttt{null}$ and $A_k = \mathsf{R}(\texttt{q.data}[h{+}k], v'_k)$ with $v'_k \neq \texttt{null}$. In the above traces, for brevity we have omitted the thread identifiers (τ_j) and event identifiers and represent each event with its label only. We use the $H(\texttt{enq(-)}, \tau, p, n, e)$ prefix to extract its specific events, e.g. $H(\texttt{enq(-)}, \tau, p, n, e).inv$.

Let us write `q.tail` to denote the index of the last entry in the queue. Observe that each `enq` operation leaves the `q.head` value unchanged while increasing `q.tail` by 1. Similarly, each `deq` operation leaves `q.tail` unchanged while increasing `q.head` by one. Note that in each $H(\texttt{enq}(v), \tau, p, n, e)$, the $e{-}1$ denotes the value of `q.tail` immediately before the insertion of node n by $H(\texttt{enq}(v), \tau, p, n, e)$, i.e. the e denotes the value of `q.tail` immediately after the insertion of node n by $H(\texttt{enq}(v), \tau, p, n, e)$. Similarly, in each $H(\texttt{deq()}, \tau, p, n, h)$, the h denotes the value of `q.head` immediately before the removal of node n by $H(\texttt{deq()}, \tau, p, n, h)$.

Let:

$$\texttt{lp}(H(o, \tau, p, n, e)) \triangleq \begin{cases} H(o, \tau, p, n, e).lin & \text{if } o{=}\texttt{enq}(v) \\ H(o, \tau, p, n, e).lin_1 & \text{if } o{=}\texttt{deq()} \text{ and } H(o, \tau, p, n, e).S_2{=}\emptyset \\ H(o, \tau, p, n, e).lin_2 & \text{if } o{=}\texttt{deq()} \text{ and } H(o, \tau, p, n, e).S_2{\neq}\emptyset \end{cases}$$

For each $\tau_j \in dom(\mathsf{P})$ let:

$$E^P_{(i,j)} = E^P_i \cap \left\{ e \,\middle|\, \texttt{tid}(e) = \tau_j \right\} \qquad E'_{(i,j)} = E^P_{(i,j)} \cup S_{(i,j)}$$

where

$$S_{(i,j)} \triangleq \left\{ \mathsf{A}(\iota, m, r) \;\middle|\; \begin{array}{l} \exists o, a, p, n, inv, e. \\ \quad inv = \mathsf{I}(\iota, m, a) = \max\left(\textsf{nvo}|_{E^P_{(i,j)} \cap I}\right) \\ \quad \wedge\, inv \in H(o, \tau_j, p, n, e) \wedge \forall r'.\ \mathsf{A}(\iota, m, r') \notin E^P_{(i,j)} \\ \quad \wedge\, \texttt{lp}(H(o, \tau_j, p, n, e)) \in E^P_{(i,j)} \\ \quad \wedge\, (m{=}\texttt{deq} \Rightarrow r{=}n) \wedge (m{=}\texttt{enq} \Rightarrow r{=}()) \end{array} \right\}$$

Let $E'_i = \bigcup_{\tau_j \in dom(\mathsf{P})} E'_{(i,j)}$. From the definition of each $E'_{(i,j)}$ and $E^P_{(i,j)}$ we then know that $E^P_i \subseteq E'_i$ and $E'_i \in \texttt{comp}(E^P_i)$. Let $T_i = \texttt{trunc}(E'_i)$.

Let C_i denote an enumeration of $\bigcup_{\tau_j \in dom(\mathsf{P})} \{H(o_j^{P_j^{i-1}+1}, \tau_j, P_j^{i-1}{+}1, n_j^{P_j^{i-1}+1}) \cdots H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}\}$ that respects *memory order (in* $\textsf{tso}_i$*) of linearisation points.* That is, for all $H(o, \tau_j, p, n, e), H(o', \tau_{j'}, p', n', e')$, if $\texttt{lp}(H(o, \tau_j, p, n, e)) \xrightarrow{\textsf{tso}_i} \texttt{lp}(H(o', \tau_{j'}, p', n', e'))$, then $H(o, \tau_j, p, n, e) <_{C_i} H(o', \tau_{j'}, p', n', e')$.

When C_i is enumerated as $C_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1). \cdots . H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i})$, let us define

$$H_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1).inv . H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1).ack \\ \cdots .H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i}).inv . H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i}).ack$$

Lemma C.1. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, let for all $i \in \{1 \cdots n\}$, C_i be as defined above. Then, for all $H(o, \tau, p, n, e)$, $H(o', \tau', p', n', e')$, a, b, c, d, if $a \in H(o, \tau, p, n, e)$ and $b \in H(o', \tau', p', n', e')$, $C_i|_c = H(o, \tau, p, n, e)$, $C_i|_d = H(o', \tau', p', n', e')$ and $(a, b) \in \mathsf{hb}_i$, then either 1) $c = d$ and $(a, b) \in \mathsf{po}_i$; or 2) $c < d$.*

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, with C_i defined as above for all $i \in \{1 \cdots n\}$. Let $\mathsf{hb}_i^0 = \mathsf{po}_i \cup \mathsf{rf}_i$ and $\mathsf{hb}_i^{j+1} = \mathsf{hb}_i^0; \mathsf{hb}_i^j$ for all $j \in \mathbb{N}$. It is then straightforward to demonstrate that $\mathsf{hb}_i = \bigcup_{j \in \mathbb{N}} \mathsf{hb}_i^j$. As such, it suffices to show that for all $j \in \mathbb{N}$, $H(o, \tau, p, n, e)$, $H(o', \tau', p', n', e')$, a, b, c, d:

$$a \in H(o, \tau, p, n, e) \wedge b \in H(o', \tau', p', n', e') \wedge (a, b) \in \mathsf{hb}_i^j \\ \wedge\ C_i|_c = H(o, \tau, p, n, e) \wedge C_i|_d = H(o', \tau', p', n', e') \\ \Rightarrow (c = d \wedge (a, b) \in \mathsf{po}_i) \vee c < d$$

We thus proceed by induction on j.

Base case $j = 0$

Pick arbitrary $H(o, \tau, p, n, e)$, $H(o', \tau', p', n', e')$, a, b, c, d such that $a \in H(o, \tau, p, n, e)$ and $b \in H(o', \tau', p', n', e')$, $C_i|_c = H(o, \tau, p, n, e)$, $C_i|_d = H(o', \tau', p', n', e')$ and $(a, b) \in \mathsf{hb}_i^0$.

There are now 5 cases to consider: 1) $c = d$; or 2) $c \neq d$, $o = \mathtt{enq}(v)$ and $o' = \mathtt{enq}(v')$ for some v, v'; or 3) $c \neq d$, $o = \mathtt{enq}(v)$ and $o' = \mathtt{deq}()$ for some v; or 4) $c \neq d$, $o = \mathtt{deq}()$ and $o' = \mathtt{enq}(v')$ for some v'; or 5) $c \neq d$, $o = \mathtt{deq}()$ and $o' = \mathtt{deq}()$.

In case 1) we then know that either $(a, b) \in \mathsf{po}_i$ or $(b, a) \in \mathsf{po}_i$. In the former case the desired result holds immediately. In the latter case we then have $a \stackrel{\mathsf{hb}_i^0}{\to} b \stackrel{\mathsf{po}_i}{\to} a$, i.e $(a, a) \in \mathsf{hb}_i$, contradicting the assumption that hb_i is acyclic (Lemma E.1).

In case (2), there are two more cases to consider: i) $(a, b) \in \mathsf{po}_i$, or ii) $(a, b) \in \mathsf{rf}_i$. In case (2.i), we then know that $\mathtt{lp}(H(o, \tau, p, n, e)) \stackrel{\mathsf{po}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$. As both linearisation points are in $W \cup U$, from the PTSO-validity of G_i we also know that $\mathtt{lp}(H(o, \tau, p, n, e)) \stackrel{\mathsf{tso}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (2.ii) we know that either a) $\tau = \tau'$ or b) $\tau \neq \tau'$. In case (2.ii.a) we then have $(a, b) \in \mathsf{po}_i$ (since otherwise we would have a cyclic hb_i) and thus from t he proof of part (2.i) we have $c < d$ as required. In case (2.ii.b) we then know that $a = \mathtt{lp}(H(o, \tau, p, n, e))$, i.e. $\mathtt{loc}(a) = \mathtt{q.data}[e]$. Moreover, from the PTSO-validity of G_i and since $(a, b) \in \mathsf{rf}_i$ we know that $(a, b) \in \mathsf{tso}_i$. On the other hand, from the shape of enq traces we know that $(b, \mathtt{lp}(H(o', \tau', p', n', e'))) \in \mathsf{po}_i$ and thus from the PTSO-validity of G_i we have $(b, \mathtt{lp}(H(o', \tau', p', n', e'))) \in \mathsf{tso}_i$. We thus have $a \stackrel{\mathsf{tso}_i}{\to} b \stackrel{\mathsf{tso}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$ and thus from the transitivity of tso_i we have $a = \mathtt{lp}(H(o, \tau, p, n, e)) \stackrel{\mathsf{tso}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (3) there are two more cases to consider: i) $(a, b) \in \mathsf{po}_i$, or ii) $(a, b) \in \mathsf{rf}_i$. In case (3.i), we then know that $\mathtt{lp}(H(o, \tau, p, n, e)) \stackrel{\mathsf{po}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$. As both linearisation points are in $W \cup U$, from the PTSO-validity of G_i we also know that $\mathtt{lp}(H(o, \tau, p, n, e)) \stackrel{\mathsf{tso}_i}{\to} \mathtt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (3.ii) we know that either a) $\tau = \tau'$ or b) $\tau \neq \tau'$. In case (3.ii.a) we then have $(a, b) \in \text{po}_i$ (since otherwise we would have a cyclic hb_i) and thus from t he proof of part (3.i) we have $c < d$ as required.

In case (3.ii.b) we then know that either 1) $a = \texttt{lp}(H(o, \tau, p, n, e))$, $b = H(o', \tau', p', n', e').r$, i.e. $e = e'$; or 2) $\texttt{loc}(a) = n.\texttt{t}$ or $\texttt{loc}(a) = n.\texttt{pc}$. In case (3.ii.b.1) from the PTSO-validity of G_i and since $(a, b) \in \text{rf}_i$ we know that $(a, b) \in \text{tso}_i$. On the other hand, from the shape of deq traces we know that $(b, \texttt{lp}(H(o', \tau', p', n', e'))) \in \text{po}_i$. Thus from PTSO-validity of G_i we have $(b, \texttt{lp}(H(o', \tau', p', n', e')))$ $\in \text{tso}_i$. We thus have $a \xrightarrow{\text{tso}_i} b \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$ and thus from the transitivity of tso_i we have $a{=}\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (3.ii.b.2) from the shape of the traces we also know $\big(\texttt{lp}(H(o, \tau, p, n, e)), H(o', \tau', p', n', e').r\big)$ $\in \text{rf}_i$ and thus from the proof of part (3.ii.b.1) we have $c < d$, as required.

In case (4) there are two more cases to consider: i) $(a, b) \in \text{po}_i$, or ii) $(a, b) \in \text{rf}_i$. In case (4.i), we then know that $\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{po}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As both linearisation points are in $W \cup U$, from the PTSO-validity of G_i we also know that $\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (4.ii) we know that either a) $\tau = \tau'$ or b) $\tau \neq \tau'$. In case (4.ii.a) we then have $(a, b) \in \text{po}_i$ (since otherwise we would have a cyclic hb_i) and thus from t he proof of part (4.i) we have $c < d$ as required.

In case (4.ii.b) we then know that $a = \texttt{lp}(H(o, \tau, p, n, e))$. From the PTSO-validity of G_i and since $(a, b) \in \text{rf}_i$ we know that $(a, b) \in \text{tso}_i$. On the other hand, from the shape of enq traces we know that $(b, \texttt{lp}(H(o', \tau', p', n', e'))) \in \text{po}_i$ and thus from the PTSO-validity of G_i we have $(b, \texttt{lp}(H(o', \tau', p', n', e'))) \in \text{tso}_i$. We thus have $a \xrightarrow{\text{tso}_i} b \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$ and thus from the transitivity of tso_i we have $a{=}\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (5) there are two more cases to consider: i) $(a, b) \in \text{po}_i$, or ii) $(a, b) \in \text{rf}_i$. In case (5.i), we then know that $\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{po}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As both linearisation points are in $W \cup U$, from the PTSO-validity of G_i we also know that $\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

In case (5.ii) we know that either a) $\tau = \tau'$ or b) $\tau \neq \tau'$. In case (5.ii.a) we then have $(a, b) \in \text{po}_i$ (since otherwise we would have a cyclic hb_i) and thus from t he proof of part (5.i) we have $c < d$ as required.

In case (5.ii.b) we then know that $a = \texttt{lp}(H(o, \tau, p, n, e))$ From the PTSO-validity of G_i and since $(a, b) \in \text{rf}_i$ we know that $(a, b) \in \text{tso}_i$. On the other hand, from the shape of deq traces we know that $(b, \texttt{lp}(H(o', \tau', p', n', e'))) \in \text{po}_i$ and thus from the PTSO-validity of G_i we have $(b, \texttt{lp}(H(o', \tau', p', n', e'))) \in \text{tso}_i$. We thus have $a \xrightarrow{\text{tso}_i} b \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$ and thus from the transitivity of tso_i we have $a{=}\texttt{lp}(H(o, \tau, p, n, e)) \xrightarrow{\text{tso}_i} \texttt{lp}(H(o', \tau', p', n', e'))$. As such, from the definition of C_i we know that $c < d$, as required.

Inductive case $j = m{+}1$

$$
\begin{array}{l}
\forall j' \in \mathbb{N}.\ \forall H(o, \tau, p, n, e), H(o', \tau', p', n', e'), a, b, c, d. \\
\quad j' \leq m \wedge a \in H(o, \tau, p, n, e) \wedge b \in H(o', \tau', p', n', e') \wedge (a, b) \in \text{hb}_i^{j'} \\
\quad \wedge\ C_i^k\big|_c = H(o, \tau, p, n, e) \wedge C_i^k\big|_d = H(o', \tau', p', n', e') \\
\qquad\qquad \Rightarrow (c = d \wedge (a, b) \in \text{po}_i) \vee c < d
\end{array}
\tag{I.H.}
$$

Pick arbitrary $H(o,\tau,p,n,e)$, $H(o',\tau',p',n',e')$, a,b,c,d such that $a \in H(o,\tau,p,n,e)$ and $b \in H(o',\tau',p',n',e')$, $C_i^k\big|_c = H(o,\tau,p,n,e)$, $C_i^k\big|_d = H(o',\tau',p',n',e')$ and $(a,b) \in \mathsf{hb}_i^j$. From the definition of hb_i^j we then know there exists f such that $(a,f) \in \mathsf{hb}_i^0$ and $(f,b) \in \mathsf{hb}_m$. We thus know there exists $H(o'',\tau'',p'',n'',e'')$ and g such that $f \in H(o'',\tau'',p'',n'',e'')$ and $C_i|_g = H(o'',\tau'',p'',n'',e'')$. From the proof of the base case we then know that $(c = g \land (a,f) \in \mathsf{po}_i) \lor c < g$. Similarly, from (I.H.) we know $(g = d \land (f,b) \in \mathsf{po}_i) \lor g < d$. There are then four cases to consider: 1) $(c = g \land (a,f) \in \mathsf{po}_i)$ and $(g = d \land (f,b) \in \mathsf{po}_i)$; or 2) $(c = g \land (a,f) \in \mathsf{po}_i)$ and $g < d$; or 3) $c < g$ and $(g = d \land (f,b) \in \mathsf{po}_i)$; or 4) $c < g$ and $g < d$.

In case (1) from the transitivity of $=$ and po_i we have $c = d \land (a,b) \in \mathsf{po}_i$, as required. In case (2) since $c = g$ and $g < d$ we have $c < d$, as required. In case (3) since $c < g$ and $g = d$ we have $c < d$, as required. In case (4) from the transitivity of $<$ we have $c < d$, as required.

□

Lemma C.2. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above with $C_i = H(c_i^1,\tau_i^1,p_i^1,n_i^1,e_i^1). \cdots .H(c_i^{t_i},\tau_i^{t_i},p_i^{t_i},n_i^{t_i},e_i^{t_i})$. For all $i \in \{1 \cdots n\}$, and a, b, let $O_a^b = H(c_i^a,\tau_i^a,p_i^a,n_i^a,e_i^a).inv.H(c_i^a,\tau_i^a,p_i^a,n_i^a,e_i^a).ack. \cdots .H(c_i^b,\tau_i^b,p_i^b,n_i^b,e_i^b).inv.H(c_i^b,\tau_i^b,p_i^b,n_i^b,e_i^b).ack$.*

For all $G_i = (E_i^0, E_i^P, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$, H_i, for all Q_i^0 and for all $l \in \{0 \cdots t_i\}$, $k{=}t_i{-}l$, $E_i^k = E_i^P \setminus \bigcup_{x=k+1}^{t_i} H(c_i^x,\tau_i^x,p_i^x,n_i^x,e_i^x).E$, and Q_i^k:

$$\mathtt{getQ}(Q_i^0, O_1^k) = Q_i^k \land \mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k) \Rightarrow$$
$$\exists Q_i^t.\ \mathtt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t \land \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)$$

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$. Let H_i and C_i be as defined as above for all $i \in \{1 \cdots n\}$. Pick an arbitrary $i \in \{1 \cdots n\}$, $G_i = (E_i^0, E_i^P, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$ and H_i. We proceed by induction on l.

Base case $l = 0$, $k = t_i$
Pick arbitrary Q_i^0 and Q_i^k such that $\mathtt{getQ}(Q_i^0, O_1^k) = Q_i^k$ and $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k)$. As $k = t_i$, we have $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^P)$. As $O_{k+1}^{t_i}{=}\epsilon$, we have $\mathtt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^k$, as required.

Inductive case $0 < l \leq t_i$

$$\begin{array}{l}\forall Q.\ \forall k' > k.\ \mathtt{getQ}(Q_i^0, O_1^{k'}) = Q \land \mathsf{isQ}(\mathsf{q}, Q, \mathsf{nvo}_i, E_i^0, E_i^{k'}) \Rightarrow \\ \quad \exists Q_i^t.\ \mathtt{getQ}(Q, O_{k'+1}^{t_i}) = Q_i^t \land \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)\end{array} \tag{I.H.}$$

Pick arbitrary Q_i^0 and Q_i^k such that $\mathtt{getQ}(Q_i^0, O_1^k) = Q_i^k$ and $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k)$. We are then required to show that there exists Q_i^t such that $\mathtt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t$ and $\mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)$. We then know:

$$O_{k+1}^{t_i} = H(c_i^{k+1},\tau_i^{k+1},p_i^{k+1},n_i^{k+1},e_i^{k+1}).inv.H(c_i^{k+1},\tau_i^{k+1},p_i^{k+1},n_i^{k+1},e_i^{k+1}).ack.O_{k+2}^{t_i}$$

There are now three cases to consider: 1) there exists m such that $c_i^{k+1}{=}\mathtt{enq}(m)$ and $n_i^{k+1}{=}m$; or 2) there exists $m \neq \mathtt{null}$ such that $c_i^{k+1}{=}\mathtt{deq()}$ and $n_i^{k+1}{=}m$; or 3) $c_i^{k+1}{=}\mathtt{deq()}$ and $n_i^{k+1}{=}\mathtt{null}$.

In case (1), as $\mathtt{getQ}(Q_i^0, O_1^k) = Q_i^k$, from its definition we have $\mathtt{getQ}(Q_i^0, O_1^{k+1}) = Q_i^k.m$. Let $Q_i^{k+1} = Q_i^k.m$. Given the trace $H(c_i^{k+1},\tau_i^{k+1},p_i^{k+1},n_i^{k+1},e_i^{k+1})$, since from the PTSO-validity of G_i we have $E_i^0 \times (E_i^P \setminus E_i^0) \subseteq \mathsf{nvo}_i$ and as $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k)$ holds, from its definition we have $\mathsf{isQ}(\mathsf{q}, Q_i^{k+1}, \mathsf{nvo}_i, E_i^0, E_i^{k+1})$. From (I.H.) we know there exists Q_i^t such that $\mathtt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$ and $\mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)$. As $\mathtt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$, by definition we also have $\mathtt{getQ}(Q_i^k, O_{k+1}^{t_i})$

$= Q_i^t$, as required.

In case (2), given the trace of $H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1})$ we know that there exists w, r, a such that $w \in U$, $\texttt{loc}(w)$=$\texttt{q.data}[a]$, $\texttt{val}_\texttt{w}(w)$=$m$, $r = H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1}).r$ and $(w, r) \in \mathsf{rf}_i$. Since G_i is PTSO-valid, we know either:
i) $w \in E_i^0$ and for all $j \in \{1 \cdots k\}$ $H(c_i^j, \tau_i^j, p_i^j, n_i^j, e_i^j).E \cap (W \cup U)_{\texttt{q.data[a]}} = \emptyset$; or
ii) there exists j such that $1 \leq j \leq k$ and w=$H(c_i^j, \tau_i^j, p_i^j, n_i^j, e_i^j).lin$ and $c_i^j = \texttt{enq}(m)$.
As $E_i^0 \subseteq E_i^P$ and the events of $H(c_i^j, \tau_i^j, p_i^j, n_i^j, e_i^j)$ are persistent (discussed above in the construction of H_i), in both cases we know that $w \in E_i^k$.
It is straightforward to demonstrate that each enq operation in H_i writes to a unique index in q.data. I case (ii) we thus know for all $j' \in \{1 \cdots k\} \setminus \{j\}$, $H(c_i^{j'}, \tau_i^{j'}, p_i^{j'}, n_i^{j'}, e_i^{j'}).E \cap (W \cup U)_{\texttt{q.data[}a\texttt{]}} = \emptyset$. That is, $\max\left(\mathsf{nvo}|_{E_i^k \cap (W \cup U)_{\texttt{q.data[}a\texttt{]}}}\right) = w$. Consequently, in both cases we have $\max\left(\mathsf{nvo}|_{E_i^k \cap (W \cup U)_{\texttt{q.data[}a\texttt{]}}}\right) = w$. On the other hand, since $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k)$ holds, from its definition we know $\texttt{val}_\texttt{w}(\max\left(\mathsf{nvo}|_{E_i^k \cap (W \cup U)_{\texttt{q.data[}a\texttt{]}}}\right)) = Q_i^k\big|_0$. We thus have $Q_i^k\big|_0 = m$.
Let $Q_i^k = m.Q'$ for some Q' and let $Q_i^{k+1} = Q'$. As $\texttt{getQ}(Q_i^0, O_1^k)$ holds, from its definition we also have $\texttt{getQ}(Q_i^0, O_1^{k+1}) = Q_i^{k+1}$. Given the trace $H(c_i^{k+1}, \tau_i^{k+1}, p_i^{k+1}, n_i^{k+1}, e_i^{k+1})$, as $\mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k)$ holds, from its definition we have $\mathsf{isQ}(\mathsf{q}, Q_i^{k+1}, \mathsf{nvo}_i, E_i^0, E_i^{k+1})$. From (I.H.) we then know there exists Q_i^t such that $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$ and $\mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P)$. As $\texttt{getQ}(Q_i^{k+1}, O_{k+2}^{t_i}) = Q_i^t$, from its definition we also have $\texttt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t$, as required.
Case (3) is analogous to that of case (2) and is omitted here.

□

Corollary 3. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above. For all $G_i = (E_i^0, E_i^P, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$, H_i and for all Q_i^0:*

$$\begin{aligned} &\mathsf{isQ}(\mathsf{q}, Q_i^0, \mathsf{nvo}_i, E_i^0, E_i^0) \Rightarrow \\ &\quad \exists Q_i^t.\ \texttt{getQ}(Q_i^0, H_i) = Q_i^t \wedge \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P) \end{aligned}$$

Proof. Follows immediately from the previous lemma when $k = 0$. □

Lemma C.3. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, if $H = H_1. \cdots .H_n$ with H_i defined as above for all $i \in \{1 \cdots n\}$, then:*

$$\exists Q.\ \texttt{getQ}(\epsilon, H) = Q$$

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$, with $H = H_1. \cdots .H_n$ and H_i defined as above for all $i \in \{1 \cdots n\}$. Let $Q_1^0 = \epsilon$. By definition we then have $\mathsf{isQ}(\mathsf{q}, Q_1^0, \mathsf{nvo}_1, E_1^0, E_1^0)$. On the other hand from Corollary 3 we have:

$$\begin{aligned} &\exists Q_1^t.\ \texttt{getQ}(Q_1^0, H_1) = Q_1^t \wedge \mathsf{isQ}(\mathsf{q}, Q_1^t, \mathsf{nvo}_1, E_1^0, E_1^P) \\ &\forall Q_2^0.\ \mathsf{isQ}(\mathsf{q}, Q_2^0, \mathsf{nvo}_2, E_2^0, E_2^0) \Rightarrow \\ &\quad \exists Q_2^t.\ \texttt{getQ}(Q_2^0, H_2) = Q_2^t \wedge \mathsf{isQ}(\mathsf{q}, Q_2^t, \mathsf{nvo}_2, E_2^0, E_2^P) \\ &\cdots \\ &\forall Q_n^0.\ \mathsf{isQ}(\mathsf{q}, Q_n^0, \mathsf{nvo}_n, E_n^0, E_n^0) \Rightarrow \\ &\quad \exists Q_n^t.\ \texttt{getQ}(Q_n^0, H_n) = Q_n^t \wedge \mathsf{isQ}(\mathsf{q}, Q_n^t, \mathsf{nvo}_n, E_n^0, E_n^P) \end{aligned}$$

For all $j \in \{2 \cdots n\}$, let $Q^0_j = \texttt{getQ}(Q^0_{j-1}, H_{j-1})$. From above we then have :

$$\begin{array}{l}\exists Q^t_1, \cdots, Q^t_n. \\ \quad \texttt{getQ}(Q^0_1, H_1) = Q^t_1 \land \texttt{getQ}(Q^t_1, H_2) = Q^t_2 \land \cdots \land \texttt{getQ}(Q^t_{n-1}, H_n) = Q^t_n\end{array}$$

From its definition we thus know there exists Q^t_n such that $\texttt{getQ}(Q^0_1, H_1. \cdots .H_n) = Q^t_n$. That is, there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$, as required. □

Theorem 8. *For all client programs* P *of the queue library (comprising calls to* `enq` *and* `deq` *only) and all PTSO-valid executions* $\mathcal{E}$ *of* `start(P)`*,* $\mathcal{E}$ *is persistently linearisable.*

Proof. Pick an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$ of P. For each $i \in \{1 \cdots n\}$, construct T_i and H_i as above. It then suffices to show that:

$$\forall i \in \{1 \cdots n\}. \ \forall a, b \in T_i. \ (a, b) \in \mathsf{hb}_i \Rightarrow a <_{H_i} b \tag{36}$$

$$\text{fifo}(\epsilon, H) \text{ holds when } H = H_1. \cdots .H_n \tag{37}$$

TS. (36)
Pick arbitrary $i \in \{1 \cdots n\}$, $a, b \in T_i$ such that $(a, b) \in \mathsf{hb}_i$. We then know there exist $c, \tau, p, n, e, c', \tau', p', n', e'$ such that $a \in H(c, \tau, p, n, e)$, $b \in H(c', \tau', p', n', e')$ and either:
1) $H(c, \tau, p, n, e)$=$H(c', \tau', p', n', e')$, a=$H(c, \tau, p, n, e).inv$ and $b = H(c, \tau, p, n, e).ack$; or
2) $H(c, \tau, p, n, e)$=$H(c', \tau', p', n', e')$, a=$H(c, \tau, p, n, e).ack$ and $b = H(c, \tau, p, n, e).inv$; or
3) $H(c, \tau, p, n, e) \neq H(c', \tau', p', n', e')$.

In case (1) the desired result holds immediately. In case (2) we have $b \xrightarrow{\mathsf{po}_i} a \xrightarrow{\mathsf{hb}_i} b$, and since $\mathsf{po}_i \subseteq \mathsf{hb}_i$ we have $b \xrightarrow{\mathsf{hb}_i} a \xrightarrow{\mathsf{hb}_i} b$. Consequently, from the transitivity of hb_i we have $(b, b) \in \mathsf{hb}_i$, contradicting the acyclicity of hb_i in Lemma E.1. In case (3) from Lemma C.1 and the definition of H_i we have $a <_{H_i} b$, as required.

TS. (37)
From Lemma C.3 we know there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. From the definition of fifo(., .) we know fifo(ϵ, H) holds if and only if there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. As such we have fifo$(\epsilon, H)$, as required.

□

D NON-BLOCKING MICHAEL-SCOTT QUEUE LIBRARY

As before, for an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$ of P with $G_i = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$, observe that when P comprises k threads, the trace of each execution era (via `start()` or `recover()`) comprises two stages: i) the trace of the *setup* stage by the master thread τ_0 performing initialisation or recovery, prior to the call to `run(P)`; followed (in po order) by ii) the trace of each of the constituent program threads $\tau_1 \cdots \tau_k$, provided that the execution did not crash during the setup stage.

As before, thanks to the placement of the persistent fence operations (**pfence**), for each thread τ_j, we know that the set of persistent events in execution era i, namely E^P_i, contains roughly a *prefix* (in po order) of thread τ_j's trace. More concretely, for each constituent thread $\tau_j \in \{\tau_1 \cdots \tau_k\} = dom(\mathsf{P})$, there exist $P^1_j \cdots P^n_j$ such that:

1) $\mathsf{P}[\tau_j] = o^0_j; \cdots ; o^{P^1_j}_j; o^{P^1_j+1}_j; \cdots o^{P^2_j}_j; \cdots ; o^{P^{n-1}_j+1}_j; \cdots ; o^{P^n_j}_j$, comprising `enq` and `deq` operations; and

```
q.enq(v) ≜
  pc:=getPC(); t:=getTC();
  n:=newNode(v,t,pc);
  map[t][pc]:=n; pfence;
  h:=q.head;
  find: while (q.data[h] != null)
    h:=h+1;
  if (!CAS(q.data[h],null,n))
    goto find;
  pfence;

q.deq() ≜
  pc:=getPC(); t:=getTC();
  try: h:=q.head; n:=q.data[h];
  map[t][pc]:=n;
  if (n != null) {
    t':=n.t; pc':=n.pc;
    map[t'][pc']+1:=⊤;
  } pfence;
  if (n!=null) {
    if (!CAS(q.head,h,h+1))
      goto try;
    pfence;
    map[t][pc]+1:=⊤; pfence
  } return n;

rem(n) ≜
  for(t in P){
    pc:=0
    while(map[t][pc]!=⊥){
      m:=map[t][pc];
      a:=map[t][pc]+1;
      if(n==m && a==⊤) return 1;
      pc++;
    }
  }
  return 0;

recover() ≜
  if (q==null || map==null)
    goto start();
  for(t in P) enq[t]:=-1;
  for(t in P) {
    (pc,n,a):=getProg(t);
    if (pc>=0 && isDeq(P[t][pc])) {
      if (n==null)
        P'[t]:=sub(P[t],pc+1);
      else {
        if (a==⊤)
          P'[t]:=sub(P[t],pc+1);
        else if (inIn(q,n) || rem(n))
          P'[t]:=sub(P[t],pc);
        else {
          P'[t]:=sub(P[t],pc+1);
          map[t][pc]+1:=⊤}
        t':=n.t; pc':=n.pc;
        enq[t']:=max(enq[t'],pc'+1);}
    } else if (pc<0) P'[t]:=P[t]; }
  for(t in P) {
    (pc,n,a):=getProg(t);
    if (pc>=0 && isEnq(P[t][pc])) {
      if (pc < enq[t])
        P'[t]:=sub(P[t],enq[t]);
      else if (a==⊤ || isIn(q,n))
        P'[t]:=sub(P[t],pc+1);
      else
        P'[t]:=sub(P[t],pc); }
  } pfence;
  run(P');

getProg(t) ≜
  pc:=-1; n:=⊥; a:=⊥;
  while (map[t][pc+1] != ⊥) pc++;
  if (pc>=0) {
    n:=map[t][pc]; a:=map[t][pc]+1;
  } return (pc,n,a);
```

Fig. 8. A non-blocking persistent Michael-Scott queue implementation with persistence code in blue

2) at the beginning of each execution era $i \in \{1 \cdots n\}$, the program executed by thread τ_j (calculated in `P'` and subsequently executed by calling `run(P')`) is that of `sub(P[`τ_j`],`P_j^{i-1}`+1)`, where $P_j^0 = -1$, for all j; and

3) in each execution era $i \in \{1 \cdots n\}$, the trace $H_{(i,j)}$ of each constituent thread $\tau_j \in dom(\mathsf{P})$ is of the following form:

$$\begin{aligned}
H_{(i,j)} \triangleq\ & H(o_j^{P_j^{i-1}+1}, \tau_j, P_j^{i-1}{+}1, n_j^{P_j^{i-1}+1}, e_j^{P_j^{i-1}+1}) \\
& \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i}) \\
& \xrightarrow{\text{po}} H(o_j^{P_j^i+1}, \tau_j, P_j^i{+}1, n_j^{P_j^i+1}, e_j^{P_j^i+1}) \\
& \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{m_j^i-1}, \tau_j, m_j^i{-}1, n_j^{m_j^i-1}, e_j^{m_j^i-1}) \\
& \xrightarrow{\text{po}} H'(o_j^{m_j^i}, \tau_j, m_j^i, n_j^{m_j^i}, e_j^{m_j^i})
\end{aligned}$$

for some $m_j^i, n_j^{P_j^{i-1}+1}, \cdots, n_j^{P_j^i}, n_j^{P_j^i+1}, \cdots, n_j^{m_j^i}, e_j^{P_j^{i-1}+1}, \cdots, e_j^{P_j^i}, e_j^{P_j^i+1}, \cdots, e_j^{m_j^i}$ where:

- The first two lines denote the execution of the $(P_j^{i-1}{+}1)^{\text{st}}$ to $(P_j^i)^{\text{th}}$ library calls of thread τ_j, with $H(o, \tau, p, n, e)$ defined shortly. Moreover, before crashing and proceeding to the next era, *all* volatile events (those in *PE*) in $H(o_j^{P_j^{i-1}+1}, \cdots) \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} H(o_j^{P_j^i-1}, \cdots)$ have persisted, and a *prefix* (in po order) of the volatile events in $H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i})$ have persisted. Note that this prefix may be equal to $H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i}, e_j^{P_j^i})$, in which case all its events have persisted.
- The next two lines denote the execution of the subsequent library calls of thread τ_j where $m_j^i \leq P_j^n$, with *none* of their volatile events having persisted.
- The last line denotes the execution of the $(m_j^i)^{\text{th}}$ call of thread τ_j ($m_j^i \leq P_j^n$), during which the program crashed and thus the execution of era i ended. As before, the $H'(o, \tau, p, n, e)$ denotes a (potentially full) prefix of $H(o, \tau, p, n, e)$.

The trace $H(o, \tau, p, n, e)$ of each library call is defined as follows:

$$\begin{aligned}
H(\texttt{deq()}, \tau, p, n, h) \triangleq\ & \mathit{inv}{=}\mathtt{I}(\iota_p, \texttt{deq}, ()) \xrightarrow{\text{po}} \mathsf{R}(pc, p) \xrightarrow{\text{po}} \mathsf{R}(\texttt{tid}_\tau, \tau) \xrightarrow{\text{po}} \mathit{FE} \\
& \xrightarrow{\text{po}} r_h{=}\mathsf{R}(\texttt{q.head}, h) \xrightarrow{\text{po}} r{=}\mathsf{R}(\texttt{q.data[}h\texttt{]}, n) \\
& \xrightarrow{\text{po}} \mathit{lin}_1{=}\mathsf{W}(\texttt{map[}\tau\texttt{][}p\texttt{]}, n) \xrightarrow{\text{po}} S_1 \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} S_2 \\
& \xrightarrow{\text{po}} \mathit{ack}{=}\mathsf{A}(\iota_p, \texttt{deq}, n)
\end{aligned}$$

where *FE* denotes the sequence of events, attempting but failing to set the `rem` field of the head node, with

$$S_1 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ \mathsf{R}(n\texttt{.t}, \tau') \xrightarrow{\text{po}} \mathsf{R}(n\texttt{.pc}, p') \xrightarrow{\text{po}} \mathsf{W}(\texttt{map[}\tau'\texttt{][}p'\texttt{]+1}, \top) & \text{otherwise} \end{cases}$$

$$S_2 = \begin{cases} \emptyset & \text{if } n = \texttt{null} \\ \mathit{lin}_2{=}\mathsf{W}(\texttt{q.head}, h{+}1) \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} c{=}\mathsf{W}(\texttt{map[}\tau\texttt{][}p\texttt{]+1}, \top) \xrightarrow{\text{po}} \mathsf{PF} & \text{otherwise} \end{cases}$$

for some τ', p'; and

$$\begin{aligned}
H(\texttt{enq(}v\texttt{)}, \tau, p, n, e) \triangleq\ & \mathit{inv}{=}\mathtt{I}(\iota_p, \texttt{enq}, n) \xrightarrow{\text{po}} \mathsf{R}(pc, p) \xrightarrow{\text{po}} \mathsf{R}(\texttt{tid}_\tau, \tau) \\
& \xrightarrow{\text{po}} \mathsf{W}(n\texttt{.val}, v) \xrightarrow{\text{po}} \mathsf{W}(n\texttt{.tid}, \tau) \xrightarrow{\text{po}} \mathsf{W}(n\texttt{.pc}, p) \\
& \xrightarrow{\text{po}} \mathsf{W}(\texttt{map[}\tau\texttt{][}p\texttt{]}, n) \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} \mathsf{R}(\texttt{q.head}, h) \\
& \underbrace{\xrightarrow{\text{po}} \mathsf{R}(\texttt{q.data[}h\texttt{]}, v_0) \xrightarrow{\text{po}} A_0 \cdots \mathsf{R}(\texttt{q.data[}h{+}s{-}1\texttt{]}, v_{s-1}) \xrightarrow{\text{po}} A_{s-1}}_{s \text{ times}} \\
& \xrightarrow{\text{po}} \mathsf{R}(\texttt{q.data[}h{+}s\texttt{]}, \texttt{null}) \xrightarrow{\text{po}} \mathit{lin}{=}\mathsf{U}(\texttt{q.data[}h{+}s\texttt{]}, \texttt{null}, n) \\
& \xrightarrow{\text{po}} \mathsf{PF} \xrightarrow{\text{po}} \mathit{ack}{=}\mathsf{A}(\iota_p, \texttt{enq}, ())
\end{aligned}$$

for some $s \geq 0$ such that $h{+}s = e$, and for all $k \in \{0 \cdots s{-}1\}$, either 1) $v_k \neq \texttt{null}$ and $A_k = \emptyset$; or $v_k = \texttt{null}$ and $A_k = \mathsf{R}(\texttt{q.data}[h{+}k], v'_k)$ with $v'_k \neq \texttt{null}$. In the above traces, for brevity we have omitted the thread identifiers (τ_j) and event identifiers and represent each event with its label only. We use the $H(\texttt{enq(-)}, \tau, p, n, e)$ prefix to extract its specific events, e.g. $H(\texttt{enq(-)}, \tau, p, n, e).inv$.

Let us write q.tail to denote the index of the last entry in the queue. Observe that each enq operation leaves the q.head value unchanged while increasing q.tail by 1. Similarly, each deq operation leaves q.tail unchanged while increasing q.head by one. Note that in each $H(\texttt{enq}(v), \tau, p, n, e)$, the $e{-}1$ denotes the value of q.tail immediately before the insertion of node n by $H(\texttt{enq}(v), \tau, p, n, e)$, i.e. the e denotes the value of q.tail immediately after the insertion of node n by $H(\texttt{enq}(v), \tau, p, n, e)$. Similarly, in each $H(\texttt{deq()}, \tau, p, n, h)$, the h denotes the value of q.head immediately before the removal of node n by $H(\texttt{deq()}, \tau, p, n, h)$.

Let:

$$\texttt{lp}(H(o, \tau, p, n, e)) \triangleq \begin{cases} H(o, \tau, p, n, e).lin & \text{if } o{=}\texttt{enq}(v) \\ H(o, \tau, p, n, e).lin_1 & \text{if } o{=}\texttt{deq()} \text{ and } H(o, \tau, p, n, e).S_2{=}\emptyset \\ H(o, \tau, p, n, e).lin_2 & \text{if } o{=}\texttt{deq()} \text{ and } H(o, \tau, p, n, e).S_2{\neq}\emptyset \end{cases}$$

For each $\tau_j \in dom(\mathsf{P})$ let:

$$E^P_{(i,j)} = E^P_i \cap \left\{ e \;\middle|\; \texttt{tid}(e) = \tau_j \right\} \qquad E'_{(i,j)} = E^P_{(i,j)} \cup S_{(i,j)}$$

where

$$\begin{aligned} S_{(i,j)} \triangleq & \left\{ \mathsf{A}(\iota, \textsf{enq}, ()) \;\middle|\; \begin{array}{l} \exists o, p, n, inv, e. \\ \quad inv = \mathtt{I}(\iota, \textsf{enq}, n) = \max\left(\textsf{nvo}|_{E^P_{(i,j)} \cap I}\right) \\ \quad \wedge\, inv \in H(o, \tau_j, p, n, e) \wedge \forall r'.\, \mathsf{A}(\iota, \textsf{enq}, r') \notin E^P_{(i,j)} \\ \quad \wedge\, \texttt{lp}(H(o, \tau_j, p, n, e)) \in E^P_{(i,j)} \end{array} \right\} \\ & \cup \left\{ \mathsf{A}(\iota, \textsf{deq}, n) \;\middle|\; \begin{array}{l} \exists o, p, inv, e. \\ \quad inv = \mathtt{I}(\iota, \textsf{deq}, ()) = \max\left(\textsf{nvo}|_{E^P_{(i,j)} \cap I}\right) \\ \quad \wedge\, inv \in H(o, \tau_j, p, n, e) \wedge \forall r'.\, \mathsf{A}(\iota, \textsf{deq}, r') \notin E^P_{(i,j)} \\ \quad \wedge\, \texttt{lp}(H(o, \tau_j, p, n, e)) \in E^P_{(i,j)} \wedge (n \neq \texttt{null} \Rightarrow H(o, \tau_j, p, n, e).c \in E^P_{(i,j)}) \end{array} \right\} \\ & \cup \left\{ \mathsf{A}(\iota, \textsf{deq}, n) \;\middle|\; \begin{array}{l} n \neq \texttt{null} \wedge \exists o, p, inv, e. \\ \quad inv = \mathtt{I}(\iota, \textsf{deq}, ()) = \max\left(\textsf{nvo}|_{E^P_{(i,j)} \cap I}\right) \\ \quad \wedge\, inv \in H(o, \tau_j, p, n, e) \wedge \forall r'.\, \mathsf{A}(\iota, \textsf{deq}, r') \notin E^P_{(i,j)} \\ \quad \wedge\, H(o, \tau_j, p, n, e).lin_1 \in E^P_{(i,j)} \\ \quad \wedge \forall k < j.\, \forall p', e'.\, H(\texttt{deq()}, \tau_k, p', n, e').lin_1 \notin E^P_{(i,k)} \\ \quad \wedge \exists k, p', e'.\, k \geq j \wedge H(\texttt{deq()}, \tau_k, p', n, e').lin_2 \in E^P_{(i,k)} \\ \qquad \wedge H(\texttt{deq()}, \tau_k, p', n, e').c \notin E^P_{(i,k)}) \end{array} \right\} \end{aligned}$$

Let $E'_i = \bigcup_{\tau_j \in dom(\mathsf{P})} E'_{(i,j)}$. From the definition of each $E'_{(i,j)}$ and $E^P_{(i,j)}$ we then know that $E^P_i \subseteq E'_i$ and $E'_i \in \texttt{comp}(E^P_i)$. Let $T_i = \texttt{trunc}(E'_i)$.

Let C_i denote an enumeration of $\bigcup_{\tau_j \in dom(\mathsf{P})} \{H(o_j^{P_j^{i-1}+1}, \tau_j, P_j^{i-1}{+}1, n_j^{P_j^{i-1}+1}) \cdots H(o_j^{P_j^i}, \tau_j, P_j^i, n_j^{P_j^i})\}$ that respects *memory order (in* $\textsf{tso}_i$*) of linearisation points*. That is, for all $H(o, \tau_j, p, n, e), H(o', \tau_{j'}, p', n', e')$, if $\texttt{lp}(H(o, \tau_j, p, n, e)) \xrightarrow{\textsf{tso}_i} \texttt{lp}(H(o', \tau_{j'}, p', n', e'))$, then $H(o, \tau_j, p, n, e) <_{C_i} H(o', \tau_{j'}, p', n', e')$.

When C_i is enumerated as $C_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1). \cdots .\, H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i})$, let us define

$$\begin{array}{l} H_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1).inv \;.\; H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1).ack \\ \quad . \cdots .H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i}).inv \;.\; H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i}).ack \end{array}$$

Lemma D.1. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, let for all $i \in \{1 \cdots n\}$, C_i be as defined above. Then, for all $H(o, \tau, p, n, e)$, $H(o', \tau', p', n', e')$, a, b, c, d, if $a \in H(o, \tau, p, n, e)$ and $b \in H(o', \tau', p', n', e')$, $C_i|_c = H(o, \tau, p, n, e)$, $C_i|_d = H(o', \tau', p', n', e')$ and $(a, b) \in \mathsf{hb}_i$, then either 1) $c = d$ and $(a, b) \in \mathsf{po}_i$; or 2) $c < d$.*

Proof. The proof of this lemma is analogous to he proof of its counterpart lemma (Lemma C.1) for the blocking MS queue implementation and is omitted here.

Lemma D.2. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above with $C_i = H(c_i^1, \tau_i^1, p_i^1, n_i^1, e_i^1). \cdots .H(c_i^{t_i}, \tau_i^{t_i}, p_i^{t_i}, n_i^{t_i}, e_i^{t_i})$. For all $i \in \{1 \cdots n\}$, and a, b, let $O_a^b = H(c_i^a, \tau_i^a, p_i^a, n_i^a, e_i^a).inv.H(c_i^a, \tau_i^a, p_i^a, n_i^a, e_i^a).ack. \cdots .H(c_i^b, \tau_i^b, p_i^b, n_i^b, e_i^b).inv.H(c_i^b, \tau_i^b, p_i^b, n_i^b, e_i^b).ack$.*

For all $G_i = (E_i^0, E_i^P, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$, H_i, for all Q_i^0 and for all $l \in \{0 \cdots t_i\}$, $k{=}t_i{-}l$, $E_i^k = E_i^P \setminus \bigcup_{x=k+1}^{t_i} H(c_i^x, \tau_i^x, p_i^x, n_i^x, e_i^x).E$, and Q_i^k:

$$\begin{array}{l} \mathtt{getQ}(Q_i^0, O_1^k) = Q_i^k \wedge \mathsf{isQ}(\mathsf{q}, Q_i^k, \mathsf{nvo}_i, E_i^0, E_i^k) \Rightarrow \\ \quad \exists Q_i^t.\ \mathtt{getQ}(Q_i^k, O_{k+1}^{t_i}) = Q_i^t \wedge \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P) \end{array}$$

Proof. The proof of this lemma is analogous to he proof of its counterpart lemma (Lemma C.2) for the blocking MS queue implementation and is omitted here.

Corollary 4. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, let for all $i \in \{1 \cdots n\}$, H_i be defined as above. For all $G_i = (E_i^0, E_i^P, E_i, \mathsf{po}_i, \mathsf{rf}_i, \mathsf{tso}_i, \mathsf{nvo}_i)$, H_i and for all Q_i^0:*

$$\begin{array}{l} \mathsf{isQ}(\mathsf{q}, Q_i^0, \mathsf{nvo}_i, E_i^0, E_i^0) \Rightarrow \\ \quad \exists Q_i^t.\ \mathtt{getQ}(Q_i^0, H_i) = Q_i^t \wedge \mathsf{isQ}(\mathsf{q}, Q_i^t, \mathsf{nvo}_i, E_i^0, E_i^P) \end{array}$$

Proof. Follows immediately from the previous lemma when $k = 0$. □

Lemma D.3. *Given a PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, if $H = H_1. \cdots .H_n$ with H_i defined as above for all $i \in \{1 \cdots n\}$, then:*

$$\exists Q.\ \mathtt{getQ}(\epsilon, H) = Q$$

Proof. Pick an arbitrary PTSO-valid execution $\mathcal{E} = G_1; \cdots ; G_n$, with $H = H_1. \cdots .H_n$ and H_i defined as above for all $i \in \{1 \cdots n\}$. Let $Q_1^0 = \epsilon$. By definition we then have $\mathsf{isQ}(\mathsf{q}, Q_1^0, \mathsf{nvo}_1, E_1^0, E_1^0)$. On the other hand from Corollary 4 we have:

$$\begin{array}{l} \exists Q_1^t.\ \mathtt{getQ}(Q_1^0, H_1) = Q_1^t \wedge \mathsf{isQ}(\mathsf{q}, Q_1^t, \mathsf{nvo}_1, E_1^0, E_1^P) \\ \forall Q_2^0.\ \mathsf{isQ}(\mathsf{q}, Q_2^0, \mathsf{nvo}_2, E_2^0, E_2^0) \Rightarrow \\ \quad \exists Q_2^t.\ \mathtt{getQ}(Q_2^0, H_2) = Q_2^t \wedge \mathsf{isQ}(\mathsf{q}, Q_2^t, \mathsf{nvo}_2, E_2^0, E_2^P) \\ \ldots \\ \forall Q_n^0.\ \mathsf{isQ}(\mathsf{q}, Q_n^0, \mathsf{nvo}_n, E_n^0, E_n^0) \Rightarrow \\ \quad \exists Q_n^t.\ \mathtt{getQ}(Q_n^0, H_n) = Q_n^t \wedge \mathsf{isQ}(\mathsf{q}, Q_n^t, \mathsf{nvo}_n, E_n^0, E_n^P) \end{array}$$

For all $j \in \{2 \cdots n\}$, let $Q_j^0 = \mathtt{getQ}(Q_{j-1}^0, H_{j-1})$. From above we then have :

$$\begin{array}{l} \exists Q_1^t, \cdots, Q_n^t. \\ \quad \mathtt{getQ}(Q_1^0, H_1) = Q_1^t \wedge \mathtt{getQ}(Q_1^t, H_2) = Q_2^t \wedge \cdots \wedge \mathtt{getQ}(Q_{n-1}^t, H_n) = Q_n^t \end{array}$$

From its definition we thus know there exists Q_n^t such that $\texttt{getQ}(Q_1^0, H_1.\cdots.H_n) = Q_n^t$. That is, there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$, as required. □

Theorem 9. *For all client programs* P *of the queue library (comprising calls to* `enq` *and* `deq` *only) and all PTSO-valid executions* $\mathcal{E}$ *of* `start(P)`*,* $\mathcal{E}$ *is persistently linearisable.*

Proof. Pick an arbitrary program P and a PTSO-valid execution $\mathcal{E} = G_1; \cdots; G_n$ of P. For each $i \in \{1 \cdots n\}$, construct T_i and H_i as above. It then suffices to show that:

$$\forall i \in \{1 \cdots n\}.\ \forall a, b \in T_i.\ (a, b) \in \mathsf{hb}_i \Rightarrow a \prec_{H_i} b \tag{38}$$

$$\text{fifo}(\epsilon, H) \text{ holds when } H = H_1.\cdots.H_n \tag{39}$$

TS. (38)
Pick arbitrary $i \in \{1 \cdots n\}$, $a, b \in T_i$ such that $(a, b) \in \mathsf{hb}_i$. We then know there exist $c, \tau, p, n, e, c', \tau', p', n', e'$ such that $a \in H(c, \tau, p, n, e)$, $b \in H(c', \tau', p', n', e')$ and either:
1) $H(c, \tau, p, n, e) = H(c', \tau', p', n', e')$, $a = H(c, \tau, p, n, e).inv$ and $b = H(c, \tau, p, n, e).ack$; or
2) $H(c, \tau, p, n, e) = H(c', \tau', p', n', e')$, $a = H(c, \tau, p, n, e).ack$ and $b = H(c, \tau, p, n, e).inv$; or
3) $H(c, \tau, p, n, e) \neq H(c', \tau', p', n', e')$.

In case (1) the desired result holds immediately. In case (2) we have $b \xrightarrow{\mathsf{po}_i} a \xrightarrow{\mathsf{hb}_i} b$, and since $\mathsf{po}_i \subseteq \mathsf{hb}_i$ we have $b \xrightarrow{\mathsf{hb}_i} a \xrightarrow{\mathsf{hb}_i} b$. Consequently, from the transitivity of hb_i we have $(b, b) \in \mathsf{hb}_i$, contradicting the acyclicity of hb_i in Lemma E.1. In case (3) from Lemma D.1 and the definition of H_i we have $a \prec_{H_i} b$, as required.

TS. (39)
From Lemma D.3 we know there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. From the definition of fifo(., .) we know $\text{fifo}(\epsilon, H)$ holds if and only if there exists Q such that $\texttt{getQ}(\epsilon, H) = Q$. As such we have $\text{fifo}(\epsilon, H)$, as required.

□

E AUXILIARY RESULTS

Lemma E.1. *For all PTSO-valid execution graphs* $G = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$*, then* $\mathsf{acyclic}(\mathsf{hb})$ *holds, where* $\mathsf{hb} = (\mathsf{po} \cup \mathsf{rf})^+$.

Proof. We proceed by contradiction. Let us assume that $\mathsf{acyclic}(\mathsf{hb})$ does not hold and there exists a such that $(a, a) \in \mathsf{hb}$. From Lemma E.2 below we then have $(a, a) \in \mathsf{po} \cup \mathsf{tso}$. That is, either: 1) $(a, a) \in \mathsf{po}$; or 2) $(a, a) \in \mathsf{tso}$. However, both cases lead to a contradiction as since G is valid, we know both po and tso are strict orders. □

Lemma E.2. *For all PTSO-valid execution graphs* $G = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$ *and for all* a, b*, if* $(a, b) \in \mathsf{hb} = (\mathsf{po} \cup \mathsf{rf})^+$*, then* $(b, a) \in \mathsf{po} \cup \mathsf{tso}$.

Proof. Pick an arbitrary PTSO-valid execution graph $G = (E^0, E^P, E, \mathsf{po}, \mathsf{rf}, \mathsf{tso}, \mathsf{nvo})$. Note that $\mathsf{hb} = (\mathsf{po} \cup \mathsf{rf})^+ = (\mathsf{po} \cup (\mathsf{rf} \setminus \mathsf{po}))^+$. Let $\mathsf{hb}_0 = \mathsf{po} \cup (\mathsf{rf} \setminus \mathsf{po})$ and $\mathsf{hb}_{i+1} = \mathsf{hb}_0; \mathsf{hb}_i$, for all $i \in \mathbb{N}$. As hb is a transitive closure, it is straightforward to demonstrate that $\mathsf{hb} = \bigcup_{i \in \mathbb{N}} \mathsf{hb}_i$. We thus show instead that:

$$\forall i \in \mathbb{N}.\ \forall a, b.\ (a, b) \in \mathsf{hb}_i \Rightarrow (a, b) \in \mathsf{po} \cup \mathsf{tso}$$

We proceed by induction on i.

Base case $i = 0$

Pick an arbitrary a, b such that $(a, b) \in \mathsf{hb}_0$. There are two cases to consider: either $(a, b) \in \mathsf{po}$, or $(a, b) \in \mathsf{rf} \setminus \mathsf{po}$. In the former case the desired result holds immediately. In the latter case, as from the PTSO-validity of G we know $\mathsf{rf} \subseteq \mathsf{tso} \cup \mathsf{po}$ and as $(a, b) \in \mathsf{rf} \setminus \mathsf{po}$, we know that $(a, b) \in \mathsf{tso}$, as required.

Inductive case $i = n+1$

$$\forall j \in \mathbb{N}.\ \forall a, b.\ j \leq n \wedge (a, b) \in \mathsf{hb}_j \Rightarrow (a, b) \in \mathsf{po} \cup \mathsf{tso} \qquad \text{(I.H.)}$$

Pick an arbitrary a, b such that $(a, b) \in \mathsf{hb}_i$. From the definition of hb_i we then know there exists c such that $(a, c) \in \mathsf{po} \cup (\mathsf{rf} \setminus \mathsf{po})$ and $(c, b) \in \mathsf{hb}_n$.

There are two cases to consider: either 1) $(a, c) \in \mathsf{po}$; or 2) $(a, c) \in \mathsf{rf} \setminus \mathsf{po}$.

In case (1), let $\mathsf{hb}_{-1} = \mathsf{id}$. From the definition of hb_n we then know there exists d such that $(c, d) \in \mathsf{po} \cup (\mathsf{rf} \setminus \mathsf{po})$ and $(d, b) \in \mathsf{hb}_{n-1}$. There are two more cases to consider: i) $(c, d) \in \mathsf{po}$; or ii) $(c, d) \in \mathsf{rf} \setminus \mathsf{po}$.

In case (1.i) we have $a \xrightarrow{\mathsf{po}} c \xrightarrow{\mathsf{po}} d$ and thus from the transitivity of po we have $(a, d) \in \mathsf{po} \subseteq \mathsf{hb}_0$. As $(d, b) \in \mathsf{hb}_{n-1}$, from the definition of hb_n we have $(a, b) \in \mathsf{hb}_n$. Consequently, from (I.H.) we have $(a, b) \in \mathsf{po} \cup \mathsf{tso}$, as required.

In case (1.ii), from the PTSO-validity of G we know $\mathsf{rf} \subseteq \mathsf{tso} \cup \mathsf{po}$. Since $(c, d) \in \mathsf{rf} \setminus \mathsf{po}$, we thus know that $(c, d) \in \mathsf{tso}$. On the other hand, from the validity of G we know $\mathsf{po} \setminus (W \times R) \subseteq \mathsf{tso}$. Moreover, as $(c, d) \in \mathsf{rf}$, we know that $c \in W$. As $(a, c) \in \mathsf{po}$ and $c \in W$, we thus have $(a, c) \in \mathsf{tso}$. We then have $a \xrightarrow{\mathsf{tso}} c \xrightarrow{\mathsf{tso}} d$, and thus from the transitivity of tso we have $(a, d) \in \mathsf{tso}$. There are now to cases to consider: a) $n = 0$ and thus $\mathsf{hb}_{n-1} = \mathsf{id}$; or b) $n > 0$.

In case (1.ii.a), as $(d, b) \in \mathsf{hb}_{n-1} = \mathsf{id}$, we have $d = b$ and thus $(a, b) \in \mathsf{tso}$, as required.

In case (1.ii.b), since $(d, b) \in \mathsf{hb}_{n-1}$, from (I.H.) we have $(d, b) \in \mathsf{po} \cup \mathsf{tso}$. On the other hand, from the validity of G we know $\mathsf{po} \setminus (W \times R) \subseteq \mathsf{tso}$. Moreover, as $(c, d) \in \mathsf{rf}$, we know that $d \in R$. As such, we have $(d, b) \in \mathsf{tso}$. We then have $a \xrightarrow{\mathsf{tso}} d \xrightarrow{\mathsf{tso}} b$, and thus from the transitivity of tso we have $(a, b) \in \mathsf{tso}$, as required.

In case (2), from the PTSO-validity of G we know $\mathsf{rf} \subseteq \mathsf{tso} \cup \mathsf{po}$. Since $(a, c) \in \mathsf{rf} \setminus \mathsf{po}$, we thus know that $(a, c) \in \mathsf{tso}$. On the other hand, since$(c, b) \in \mathsf{hb}_n$, from (I.H.) we have $(c, b) \in \mathsf{po} \cup \mathsf{tso}$. There are two more cases to consider: i) $(c, b) \in \mathsf{tso}$; or ii) $(c, b) \in \mathsf{po}$.

In case (2.i) we have $a \xrightarrow{\mathsf{tso}} c \xrightarrow{\mathsf{tso}} b$, and thus from the transitivity of tso we have $(a, b) \in \mathsf{tso}$, as required.

In case (2.ii), from the validity of G we know $\mathsf{po} \setminus (W \times R) \subseteq \mathsf{tso}$. On the other hand, since $(a, c) \in \mathsf{rf}$, we know that $c \in R$. As such, we have $(c, b) \in \mathsf{tso}$. We thus have $a \xrightarrow{\mathsf{tso}} c \xrightarrow{\mathsf{tso}} b$, and thus from the transitivity of tso we have $(a, b) \in \mathsf{tso}$, as required.

□