$$[\![.]\!]_{\text{PSER}} : \textsc{Com}_{\text{PSER}} \to (\textsc{TId} \times \mathbb{N}^+ \times \textsc{Store}) \to \textsc{RV} \qquad \{\!|.|\!\}_{\text{PSER}} : \textsc{Prog}_{\text{PSER}} \to \textsc{RV}$$

$$\begin{aligned}[\![\mathtt{C}]\!]_{\text{PSER}}(\tau,n,\mathrm{s}) &\triangleq \left\{\langle v, \mathtt{pserG}(G,\tau,n)\rangle \;\middle|\; v \in \textsc{Val} \wedge \langle v, G\rangle \in [\![\mathtt{C}]\!](\mathrm{s})\right\}\\ &\quad\cup \left\{\langle \bot, \mathtt{pserPG}(G,\tau,n)\rangle \;\middle|\; \langle \bot, G\rangle \in [\![\mathtt{C}]\!](\mathrm{s})\right\}\end{aligned}$$

$$[\![\mathtt{C};\mathtt{C}_{\text{PSER}}]\!]_{\text{PSER}}(\tau,n,\mathrm{s}) \triangleq \left\{\langle r_2, G_1; G_2\rangle \;\middle|\; \begin{array}{l}\langle v_1, G_1\rangle \in [\![\mathtt{C}]\!]_{\text{PSER}}(\tau,n,\mathrm{s})\\ \wedge\ \langle r_2, G_2\rangle \in [\![\mathtt{C}_{\text{PSER}}]\!]_{\text{PSER}}(\tau,n{+}1,\mathrm{s})\end{array}\right\}$$
$$\cup \left\{\langle r_1, G_1\rangle \;\middle|\; \langle r_1, G_1\rangle \in [\![\mathtt{C}]\!]_{\text{PSER}}(\tau,n,\mathrm{s}) \wedge \nexists v.\ r_1 = v\right\}$$

$$\{\!|\mathtt{C}_1 \| \cdots \| \mathtt{C}_n|\!\}_{\text{PSER}} \triangleq \left\{\mathsf{par}(r_1, G_1, \cdots, r_n, G_n) \;\middle|\; \forall i \le n.\ \langle r_i, G_i\rangle \in [\![\mathtt{C}_i]\!]_{\text{PSER}}(\tau_i, 1, \mathrm{s}_0)\right\}$$

$$\mathtt{pserG}(G,\tau,n) \triangleq (E', \mathrm{po}') \qquad \text{where} \qquad E'{=}\{b,e\} \uplus \left\{f(a) \;\middle|\; a \in G.E\right\}$$
with
$$f(\langle i,\tau',l\rangle) \triangleq \begin{cases}\langle i,\tau',(\mathtt{R},x,v,\langle\tau,n\rangle)\rangle & \text{if } l \in \textsc{RLab} \wedge \mathtt{loc}(l){=}x \wedge \mathtt{val_r}(l){=}v\\ \langle i,\tau',(\mathtt{W},x,v,\langle\tau,n\rangle)\rangle & \text{if } l \in \textsc{WLab} \wedge \mathtt{loc}(l){=}x \wedge \mathtt{val_r}(l){=}v\end{cases}$$

$$\mathtt{lab}(b){=}(\mathtt{B},\langle\tau,n\rangle) \qquad \text{and} \qquad \mathtt{lab}(e){=}(\mathtt{E},\langle\tau,n\rangle)$$
and
$$\mathrm{po}' = \left\{(b,a) \;\middle|\; a \in E'\right\} \cup \left\{(a,e) \;\middle|\; a \in E'\right\} \cup \left\{(f(c),f(d)) \;\middle|\; (c,d) \in G.\mathrm{po}\right\}$$

$$\mathtt{pserPG}(G,\tau,n) \triangleq (E'', \mathrm{po}'')$$
with $\quad \mathtt{lab}(b){=}(\mathtt{B},\langle\tau,n\rangle) \qquad E''{=}\{b\} \uplus \left\{f(a) \;\middle|\; a \in G.E\right\}$
$$\mathrm{po}'' = \left\{(b,a) \;\middle|\; a \in E''\right\} \cup \left\{(f(c),f(d)) \;\middle|\; (c,d) \in G.\mathrm{po}\right\}$$

Fig. 5. The semantics of PSER programs

A PSER: AUXILIARY DEFINITIONS AND THEOREMS

Definition 16 (PSER semantics). The *semantics of PSER programs* is as given in Fig. 5.

Definition 17 (Graph composition). Given two PSER executions $G_1{=}(E_1, I_1, P_1, \mathrm{po}_1, \mathrm{rf}_1, \mathrm{mo}_1, \mathrm{nvo}_1)$ and $G_2{=}\langle E_2, I_2, P_2, \mathrm{po}_2, \mathrm{rf}_2, \mathrm{mo}_2, \mathrm{nvo}_2\rangle$, whenever G_1 and G_2 contain disjoint events ($E_1 \cap E_2 = \emptyset$), then their *composition*, written $\mathtt{flat}(G_1, G_2)$, is given by $\langle E, I, P, \mathrm{po}, \mathrm{rf}, \mathrm{mo}, \mathrm{nvo}\rangle$, where $E \triangleq E_1' \cup E_2'$ with $E_1' \triangleq \left\{e \in G_1.T \;\middle|\; [e]_{\mathrm{st}} \cap D \subseteq P_1\right\}$ and $E_2' \triangleq E_2 \setminus I_2$, $P \triangleq (E_1' \cap P_1) \cup (E_2' \cap P_2)$, $I \triangleq I_1$, $\mathrm{po} \triangleq \mathrm{po}_1|_{E_1'} \cup \mathrm{po}_2|_{E_2'} \cup (E_1' \times E_2')_i \cup (I_1 \times E_2')$, and:

$$\begin{aligned}\mathrm{rf} &\triangleq \mathrm{rf}_1|_{E_1'} \cup \mathrm{rf}_2|_{E_2'} \cup \left\{(w,r) \;\middle|\; \exists w' \in I_2.\ (w',r) \in \mathrm{rf}_2 \wedge\ w = \max\left(\mathrm{nvo}_1|_{P_1 \cap W_{\mathtt{loc}(r)}}\right)\right\}\\ \mathrm{mo} &\triangleq \mathrm{mo}_1|_{E_1'} \cup \mathrm{mo}_2|_{E_2'} \cup (E_1' \cap W \times E_2' \cap W)_{loc} \cup I_1 \times E_2 \cap W_{loc}\\ \mathrm{nvo} &\triangleq \mathrm{nvo}_1|_{E_1'} \cup \mathrm{nvo}_2|_{E_2'} \cup (E_1' \cap D \times E_2' \cap D) \cup I_1 \times E_2 \cap D\end{aligned}$$

Given an execution chain $C{=}G_1, \cdots, G_n$, its *flattened execution*, written $\mathtt{flat}(C)$, is given by G_n', where $G_1' \triangleq G_1$ and $G_i' \triangleq \mathtt{flat}(G_{i-1}', G_i)$ for $i \in \{2 \cdots n\}$.

Lemma 1. *For all* $\mathbb{P}{=}\langle \mathsf{P}, \mathtt{rec}_{PSER}\rangle$, $G_1 \in \mathtt{pexec}(\mathsf{P})$, $G_2 \in \mathtt{pexec}(\mathtt{rec}_{PSER}(\mathsf{P}, G_1))$ *and* $G{=}\mathtt{flat}(G_1, G_2)$, *if* G_1 *and* G_2 *are PSER-consistent, then:*

(1) $\langle -, \langle G.E, G.\mathrm{po}\rangle\rangle \in \{\!|\mathsf{P}|\!\}_{PSER}$*; and*
(2) if $G_2 \in \mathtt{exec}(\mathtt{rec}_{PSER}(\mathsf{P}, G_1))$*, then there exists* $v \in \textsc{Val}$ *such that* $\langle v, \langle G.E, G.\mathrm{po}\rangle\rangle \in \{\!|\mathsf{P}|\!\}_{PSER}$.

Proof. Pick arbitrary P, $G_1 \in \mathtt{pexec}(\mathsf{P})$, $G_2 \in \mathtt{pexec}(\mathtt{rec}_{\mathrm{PSER}}(\mathsf{P}, G_1))$ such that G_1 and G_2 are PSER-consistent. Let $\mathbb{P}{=}\langle \mathsf{P}, \mathtt{rec}_{\mathrm{PSER}}\rangle$, $G{=}\mathtt{flat}(G_1, G_2)$ and $\mathsf{P}'{=}\mathtt{rec}_{\mathrm{PSER}}(\mathsf{P}, G_1)$.

RTS. (1)
Let us write $\xi_{i,j}$ for $\langle \tau_i, j\rangle$; given an execution G', we write $G'.\xi$ for $G'.E \cap \left\{a \;\middle|\; \mathtt{tx}(a){=}\xi\right\}$. For each $\tau_i \in dom(\mathsf{P})$, from the PSER-consistency of G_1 we know there exists m_i such that $(G_1.\xi_{i,1} \cup \cdots \cup G_1.\xi_{i,m_i}) \cap D \subseteq G_1.P \subseteq G_1.T$, $(G_1.\xi_{i,k} \cap D) \not\subseteq G_1.P$, and $G_1.\xi_{i,k} \cap G_1.P{=}\emptyset$, for all $k > m_i$. As such, when $dom(\mathsf{P}){=}\{\tau_1 \cdots \tau_n\}$, we have $G_1.P{=} \bigcup_{k=1}^{m_1} (G_1.\xi_{1,k} \cap D) \cup \cdots \cup \bigcup_{k=1}^{m_n} G_1.\xi_{n,k} \cap D$.

Consequently, from the definition of $\mathtt{rec}_{\mathrm{PSER}}$ we know $\mathsf{P}'(\tau_i){=}\mathsf{sub}(\mathsf{P}(\tau_i), m_i{+}1)$. That is, there exists M_i such that $\mathsf{P}(\tau_i){=}[\mathsf{C}_1];\cdots;[\mathsf{C}_{m_i}];\cdots;[\mathsf{C}_{M_i}]$ and $\mathsf{P}'(\tau_i){=}[\mathsf{C}_{m_{i+1}}];\cdots;[\mathsf{C}_{M_i}]$, for some $\mathsf{C}_1 \cdots \mathsf{C}_{M_i}$. As such, given the definitions of G, G_2 and the semantics of PSER programs (Fig. 5), we have $\langle -, \langle G.E, G.\mathrm{po}\rangle\rangle \subseteq \mathtt{pexec}(\mathsf{P})$, as required.

RTS. (2)
This proof is analogous to that of part (1) and is thus omitted here. □

Lemma 2. *For all* $\mathbb{P}{=}\langle \mathsf{P}, \mathtt{rec}_{PSER}\rangle$, $G_1 \in \mathtt{pexec}(\mathsf{P})$, $G_2 \in \mathtt{pexec}(\mathtt{rec}_{PSER}(\mathsf{P}, G_1))$ *and* $G{=}\mathtt{flat}(G_1, G_2)$, *if* G_1 *and* G_2 *are PSER-consistent, then:*

(1) G *is PSER-consistent and* $G \in \mathtt{pexec}(\mathsf{P})$*; and*
(2) if $G_2 \in \mathtt{exec}(\mathtt{rec}_{PSER}(\mathsf{P}, G_1))$*, then* $G \in \mathtt{exec}(\mathsf{P})$.

Proof. Pick arbitrary P, $G_1 \in \mathtt{pexec}(\mathsf{P})$, $G_2 \in \mathtt{pexec}(\mathtt{rec}_{\mathrm{PSER}}(\mathsf{P}, G_1))$ such that G_1 and G_2 are PSER-consistent. Let $\mathbb{P}{=}\langle \mathsf{P}, \mathtt{rec}_{\mathrm{PSER}}\rangle$ and $G{=}\mathtt{flat}(G_1, G_2)$.

RTS. (1)
From Lemma 1 we know $\langle -, \langle G.E, G.\mathrm{po}\rangle\rangle \in \{\!|\mathsf{P}|\!\}_{\mathrm{PSER}}$. On the other hand, from the definition of $\mathtt{flat}(.)$ we know that $G \in \textsc{Exec}$. As such, from the definition of $\mathtt{pexec}(.)$ we have $G \in \mathtt{pexec}(\mathsf{P})$.

We next show that G is PSER-consistent. Note that given the definition of G, we have $G.\mathrm{po} \subseteq G_1.\mathrm{po} \cup G_2.\mathrm{po} \cup (G_1.E \times G_2.E)$; $G.\mathrm{rf} \subseteq G_1.\mathrm{rf} \cup G_2.\mathrm{rf} \cup (G_1.E \times G_2.E)$; $G.\mathrm{mo} \subseteq G_1.\mathrm{mo} \cup G_2.\mathrm{mo} \cup (G_1.E \times G_2.E)$; $G.\mathrm{rb} \subseteq G_1.\mathrm{rb} \cup G_2.\mathrm{rb} \cup (G_1.E \times G_2.E)$; and $G.\mathrm{nvo} \subseteq G_1.\mathrm{nvo} \cup G_2.\mathrm{nvo} \cup (G_1.E \times G_2.E)$. That is, $G.\mathrm{po} \cup G.\mathrm{rf} \cup G.\mathrm{mo} \cup G.\mathrm{rb} \cap (G_2.E \times G_1.E) = \emptyset$. As such, we have $G.\mathrm{hb}_{\mathrm{ser}} \subseteq G_1.\mathrm{hb}_{\mathrm{ser}} \cup G_2.\mathrm{hb}_{\mathrm{ser}} \cup (G_1.E \times G_2.E)$; and thus $G.\mathrm{hb}_{\mathrm{ser}} \cap (G_2.E \times G_1.E){=}\emptyset$. In order to show G is PSER-consistent, we are required to show:

$$(G.\mathrm{rf} \cup G.\mathrm{mo} \cup G.\mathrm{rb}) \cap G.\mathrm{st} \subseteq G.\mathrm{po}$$
$$G.\mathrm{hb}_{\mathrm{ser}} \text{ is irreflexive}$$
$$G.\mathrm{hb}_{\mathrm{ser}} \cap (D \times D) \subseteq G.\mathrm{nvo}$$
$$dom([D]; G.\mathrm{st}; [G.P]) \subseteq G.P \subseteq G.T$$
$$\mathrm{nvo}_{\mathsf{T}} \text{ is acyclic}$$

For the first part, from construction of G we have $G.\mathrm{rf} \subseteq G_1.\mathrm{rf} \cup G_2.\mathrm{rf}$; $G.\mathrm{mo} \subseteq G_1.\mathrm{mo} \cup G_2.\mathrm{mo}$; and $G.\mathrm{rb} \subseteq G_1.\mathrm{rb} \cup G_2.\mathrm{rb}$. The desired result thus follows from PSER-consistency of G_1, G_2.

For the second part, we proceed by contradiction and assume there exists a such that $(a, a) \in G.\mathrm{hb}_{\mathrm{ser}}$. There are two cases to consider: 1) $\exists c \in G_1.E, d \in G_2.E.\ a \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} c \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} d \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} a$; or 2) $\nexists c \in G_1.E, d \in G_2.E.\ a \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} c \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} d \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} a$. In case (1) we thus have $d \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} a \overset{G.\mathrm{hb}_{\mathrm{ser}}}{\longrightarrow} c$ and thus $(d, c) \in G.\mathrm{hb}_{\mathrm{ser}} \cap (G_2.E \times G_1.E)$. This however leads to a contradiction since as described above

we have $G.\mathsf{hb}_{\mathsf{ser}} \cap (G_2.E \times G_1.E)=\emptyset$. In case (2) since $G.\mathsf{hb}_{\mathsf{ser}} \cap (G_2.E \times G_1.E)=\emptyset$, we have $(a,a) \in G_1.\mathsf{hb}_{\mathsf{ser}} \cup G_2.\mathsf{hb}_{\mathsf{ser}}$. This however contradicts our assumption that G_1, G_2 are PSER-consistent.

For the third part, pick an arbitrary $(a,b) \in G.\mathsf{hb}_{\mathsf{ser}} \cap (D \times D)$. As $G.\mathsf{hb}_{\mathsf{ser}} \subseteq G_1.\mathsf{hb}_{\mathsf{ser}} \cup G_2.\mathsf{hb}_{\mathsf{ser}} \cup (G_1.E \times G_2.E)$, we know either $(a,b) \in G_1.\mathsf{hb}_{\mathsf{ser}} \cup G_2.\mathsf{hb}_{\mathsf{ser}}$, or $(a,b) \in G_1.E \times G_2.E$. In the former case the desired result follows from the PSER-consistency of G_1, G_2. In the latter case from the construction of G we have $(a,b) \in G.\mathsf{nvo}$, as required.

For the fourth part, pick an arbitrary $a \in dom([D]; G.\mathsf{st}; [G.P])$; that is there exists b such that $a \in D$, $(a,b) \in G.\mathsf{st}$ and $b \in G.P$. Given the definition of G we then know either 1) $a, b \in G_1.E$, $b \in G_1.P$ and $(a,b) \in G_1.\mathsf{st}$; or 2) $a, b \in G_2.E$, $b \in G_1.P$ and $(a,b) \in G_1.\mathsf{st}$. In both cases, the desired result follows from the PSER-consistency of G_1 and G_2.

For the last part, we proceed by contradiction and assume there exists a such that $(a,a) \in G.\mathsf{nvo}_{\mathsf{T}}{}^{+}$. There are two cases to consider: 1) $\exists c \in G_1.E, d \in G_2.E.\ a \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} c \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} d \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} a$; or 2) $\nexists c \in G_1.E, d \in G_2.E.\ a \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} c \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} d \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} a$. In case (1) we thus have $d \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} a \overset{G.\mathsf{nvo}_{\mathsf{T}}}{\rightarrow} c$ and thus $(d,c) \in G.\mathsf{nvo}_{\mathsf{T}} \cap (G_2.E \times G_1.E)$. That is, there exists d', c' such that $(d',c') \in G.\mathsf{nvo} \cap (G_2.E \times G_1.E)$. This however leads to a contradiction since as described above we have $G.\mathsf{nvo} \cap (G_2.E \times G_1.E)=\emptyset$. In case (2) since $G.\mathsf{nvo} \cap (G_2.E \times G_1.E)=\emptyset$, we have $(a,a) \in G_1.\mathsf{nvo}_{\mathsf{T}} \cup G_2.\mathsf{nvo}_{\mathsf{T}}$; i.e. $\exists b.\ (b,b) \in G_1.\mathsf{nvo} \cup G_2.\mathsf{nvo}$. This however contradicts our assumption that G_1, G_2 are PSER-consistent.

RTS. (2)
From Lemma 1 we know there exists $v \in \textsc{Val}$ such that $\langle v, \langle G.E, G.\mathrm{po}\rangle\rangle \in \{\!|\mathsf{P}|\!\}_{\mathsf{PSER}}$. On the other hand, from the definition of $\mathtt{flat}(.)$ we know that $G \in \textsc{Exec}$. As such, from the definition of $\mathtt{exec}(.)$ we have $G \in \mathtt{exec}(\mathsf{P})$, as required. □

Corollary 1. *For all persistent programs $\mathbb{P}=\langle \mathsf{P}, \mathtt{rec}_{PSER}\rangle$ and all $C \in \mathtt{chain}(\mathbb{P})$, if C is PSER-valid, then $G=\mathtt{flat}(C)$ is PSER-consistent and $G \in \mathtt{exec}(\mathsf{P})$.*

Proof. Follows from Lemma 2 by induction on the length of C. □

Lemma 3. *For all PSER-consistent executions G, $G.\mathsf{hb}_{\mathsf{ser}} \cup G.\mathsf{nvo}_{\mathsf{T}}{}^{+}$ is irreflexive, where $G.\mathsf{hb}_{\mathsf{ser}}$ is as defined in Def. 11.*

Proof. Pick arbitrary PSER-consistent execution G. We then proceed by contradiction. Let us assume that $G.\mathsf{hb}_{\mathsf{ser}} \cup G.\mathsf{nvo}_{\mathsf{T}}{}^{+}$ is not irreflexive, i.e. there exists a, b such that $(a,b) \in G.\mathsf{hb}_{\mathsf{ser}}$ and $(b,a) \in G.\mathsf{nvo}_{\mathsf{T}}{}^{+}$. From the definition of $\mathsf{nvo}_{\mathsf{T}}{}^{+}$ we then know there exist $a', b' \in D$ such that $(a,a'), (b,b') \in G.\mathsf{st}$ and $(b',a') \in \mathsf{nvo}$. On the other hand, from the definition of $G.\mathsf{hb}_{\mathsf{ser}}$ and since $(a,a'), (b,b') \in G.\mathsf{st}$ and $(a,b) \in G.\mathsf{hb}_{\mathsf{ser}}$, we know that $(a',b') \in G.\mathsf{hb}_{\mathsf{ser}}$. As such, since G is PSER-consistent (and thus $G.\mathsf{hb}_{\mathsf{ser}} \cap (D \times D) \in G.\mathsf{nvo}$), we have $(a',b') \in G.\mathsf{nvo}$. We then have an nvo cycle: $a' \overset{G.\mathsf{nvo}}{\rightarrow} b' \overset{G.\mathsf{nvo}}{\rightarrow} a'$, contradicting the assumption that G is an execution. □

Theorem 3 (Linearisability). *Given an implementation $\mathcal{I}$ of library $\mathcal{L}$, if $\mathcal{I}$ is sequentially sound, then for all programs P: (1) $\mathtt{pser}(\mathsf{P}, \mathcal{L})$ is linearisable; and (2) $\langle \mathtt{pser}(\mathsf{P}, \mathcal{L}), \mathtt{rec}_{PSER}\rangle$ is persistently linearisable.*

Proof. Pick an arbitrary library $\mathcal{L}$, a sequentially sound implementation $\mathcal{I}$ of $\mathcal{L}$, and program P.

RTS. (1)
Pick an arbitrary $G \in \mathtt{exec}(\mathtt{pser}(\mathsf{P}, \mathcal{L}))$ such that G is PSER-consistent. Since G is a full execution and is PSER-consistent we have $G.E|_D = G.P \subseteq G.T \subseteq G.E$. As such we know $S \triangleq G.T \setminus G.E \subseteq R$

and thus $[S]; G.\mathsf{hb}_{\mathsf{ser}} \cup G.\mathsf{nvo}_{\mathsf{T}}; [G.T] = \emptyset$. It then suffices to show that there exists a sequential history of $G.T$ that is $\mathcal{L}$-legal.

Let $A \triangleq \left\{(\mathtt{tx}(a), \mathtt{tx}(b)) \;\middle|\; (a,b) \in G.\mathsf{hb}_{\mathsf{ser}} \cup G.\mathsf{nvo}_{\mathsf{T}}{}^{+}\right\}$. From Lemma 3 and the transitivity of $G.\mathsf{hb}_{\mathsf{ser}}$ and $G.\mathsf{nvo}_{\mathsf{T}}{}^{+}$ we know that A is a strict partial order. Let txo denote a total extension of A on $\left\{\xi \;\middle|\; \exists a \in G.E.\ \mathtt{tx}(a){=}\xi\right\}$. Let:

$$\mathsf{to} \triangleq \left\{(a,b) \;\middle|\; (a,b) \in \mathsf{po} \cap \mathsf{st} \vee (\mathtt{tx}(a), \mathtt{tx}(b)) \in \mathsf{txo}\right\}$$

Note that since G is PSER-consistent, we know that $G.\mathsf{rf} \cap G.\mathsf{st} \subseteq G.\mathsf{po}$. As such, given the definition of to and txo we know that to is a sequential history of G. Consequently, since $\mathcal{I}$ is sequentially sound, we know that to is $\mathcal{L}$-legal.

RTS. (2)
Pick an arbitrary $G_1, \cdots, G_n{=}\mathcal{C} \in \mathtt{chain}(\langle \mathtt{pser}(\mathsf{P}, \mathcal{L}), \mathtt{rec}_{\mathsf{PSER}}\rangle)$ such that $\mathcal{C}$ is PSER-valid. For $i \in \{1 \cdots n\}$, let $E_i \triangleq \left\{e \in G_i.T \;\middle|\; [e]_{\mathsf{st}} \cap D \subseteq G_i.P\right\}$. For each E_i let us construct to_i as in the previous part. Following similar reasoning steps as in the previous part, we know to_i linearises G_i.

Let $G{=}\mathtt{flat}(\mathcal{C}){=}G_1; \cdots; G_n$. Note that from the definition of G we have $G.E{=}(E_1 \cup \cdots \cup E_n) \setminus (G_1.I \cup \cdots \cup G_n.I)$. Let $\mathsf{to} \triangleq \mathsf{to}_1|_{G.E}; \cdots; \mathsf{to}_n|_{G.E}$. From the definition of to we then have:

$$\mathsf{to} = \mathsf{to}_1; \mathsf{to}_2|_{G_2.E \setminus G_2.I}; \cdots; \mathsf{to}_n|_{G_n.E \setminus G_n.I}$$

Moreover, from the definition of to and each to_i we then know that to is a sequential history of G. Consequently, since $\mathcal{I}$ is sequentially sound, we know that to is $\mathcal{L}$-legal, as required. □

```
r-lock(x) ≜
  start: a := xl;
         if (is-odd a)
           goto start;
         if (!CAS(xl, a, a+2))
           goto start;

r-unlock(x) ≜ FAA(xl, −2);

w-lock(x) ≜ repeat (CAS(xl, 0, 1))
```

```
can-promote(x) ≜
  start: a := xl;
         if (is-odd a)
           return false;
         if (!CAS(xl, a, a−1))
           goto start;
         repeat (xl == 1);
         return true;

w-unlock(x) ≜ xl := 0;
```

Fig. 6. MRSW lock implementation, where all reads are acquire (A) reads and all writes are release (L) writes

B SOUNDNESS OF PSER IMPLEMENTATION IN PARMv8

B.1 MSRW Lock Implementation

An implementation of MSRW locks in PARMv8 is given in Fig. 6, where all reads are acquire (A) reads, all writes are release (L) writes, and all updates are acquire-release updates (A, L).

B.2 Soundness of PSER Implementation

For an arbitrary program P and a PARMv8-valid execution chain $C = G_1; \cdots; G_n$ of P with $G_i = (E_i, I_i, P_i, \mathrm{po}_i, \mathsf{rf}_i, \mathsf{mo}_i, \mathsf{nvo}_i)$, 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 *initialisation* 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 initialisation stage.

Note that as the execution is PARMv8-valid, thanks to the placement of the persistent barrier operations ($\mathbf{DSB}_{\mathtt{full}}$), for each thread τ_j, we know that the set of persistent events in execution era i, namely 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 $p_1^j \cdots p_n^j, q_1^j \cdots q_n^j, w_1^j, \cdots, w_n^j$ such that:

(1) $\mathsf{P}[\tau_j] = \mathsf{T}_j^0; \cdots; \mathsf{T}_j^{p_1^j}; \mathsf{T}_j^{p_1^j+1}; \cdots \mathsf{T}_j^{p_2^j}; \cdots; \mathsf{T}_j^{p_{n-1}^j+1}; \cdots; \mathsf{T}_j^{p_n^j}$, where each T_j^k denotes the k^{th} transaction of thread τ_j; and $\mathsf{T}_j^{p_i^j}$ denotes the last transaction of τ_j *logged* in the i^{th} era, i.e. the i^{th} crash occurred when $\mathtt{log}[\tau_j] = \xi_j^{p_i^j}$.

(2) At the beginning of each execution era $i \in \{1 \cdots n\}$, for all j, the program executed by thread τ_j (calculated in `P'` and subsequently executed by calling `run(P')`) is that of $\mathtt{sub}(\mathsf{P}[\tau_j], q_j^i)$, such that either $q_j^i = p_j^{i-1}+1$ when $w_i^j \neq \bot$, or $q_j^i = p_j^{i-1}$ when $w_i^j = \bot$, where $p_j^0 = 0$.

(3) In each execution era $i \in \{1 \cdots n\}$, the trace of the program is of the form $\theta^p_{\mathrm{init}(i)} \xrightarrow{\mathrm{po}} (\theta_{(i,1)} || \cdots || \theta_{(i,k)})$, where $\theta^p_{\mathrm{init}(i)}$ denotes a (potentially full) prefix of $\theta_{\mathrm{init}(i)}$; $\theta_{\mathrm{init}(i)}$ denotes the execution of the initialisation or recovery mechanism defined shortly; and $\theta_{(i,j)}$ denotes the trace of the j^{th} constituent thread $\tau_j \in dom(\mathsf{P})$ and is defined as follows:

$$\theta_{(i,j)} \triangleq \begin{cases} \theta_i(\xi_j^{q_j^i}) \xrightarrow{\mathrm{po}} \cdots \xrightarrow{\mathrm{po}} \theta_i^p(\xi_j^{p_j^i}) & \text{if } \theta'_{\mathrm{init}_i} = \theta_{\mathrm{init}_i} \\ \emptyset & \text{otherwise} \end{cases}$$

More concretely, whenever $\theta^{p}_{\text{init}_i}{=}\theta_{\text{init}_i}$, i.e. no crash occurred during the execution of $\theta^{p}_{\text{init}_i}$, then $\theta_{(i,j)}$ denotes the execution of the $(q^i_j)^{\text{th}}$ to o^{th} transactions of thread τ_j, with $\theta_i(\xi)$ defined shortly. We write T^i for the set of all transactions executed in the i^{th} era.

Moreover, due to the placement of the $\mathbf{DSB}_{\texttt{full}}$ instructions, before crashing and proceeding to the next era, *all* durable events in $\theta_i(\xi_j^{q^i_j}) \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} \theta_i(\xi_j^{p^{i-1}_j})$ have persisted, and a *subset* of the durable events in $\theta_i(\xi_j^{p^i_j})$ have persisted. Note that this subset may be equal to $\theta_i(\xi_j^{p^i_j})$, in which case all its durable events have persisted.

In the very first era ($i = 1$) we have $\theta_{\text{init}(1)} = \emptyset$, and when $i > 1$, the $\theta_{\text{init}(i)}$ is of the form: $Us \xrightarrow{\text{po}} C(i,1) \xrightarrow{\text{po}} W(i,1) \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} C(i,k) \xrightarrow{\text{po}} W(i,k) \xrightarrow{\text{po}} dsb$, where Us denotes the sequence of events releasing all locks, $\texttt{lab}(dsb){=}(\texttt{DSB}, \texttt{full})$, and for all $i \in \{1 \cdots n\}$ and $j \in \{1 \cdots k\}$:

$$C(i{+}1, j) \triangleq rlog_{(i+1,j)} \xrightarrow{\text{po}} rwmap_{(i+1,j)} \xrightarrow{\text{po}} wp'_{(i+1,j)}$$

where $\texttt{lab}(rlog_{(i+1,j)}) = (\texttt{R}, \texttt{log}[\tau_j], \xi_j^{p^i_j}, -)$, $\texttt{lab}(rwmap_{(i+1,j)}) = (\texttt{R}, \texttt{ws}[\xi_j^{p^i_j}], w_j^{i+1}, -)$, $\texttt{lab}(wp'_{(i+1,j)}) = (\texttt{W}, \texttt{P'}[\tau_j], q_j^{i+1}, -)$, and when $dom(w_j^{i+1}) = x_1 \cdots x_m$:

$$W(i{+}1, j) \triangleq W_1^{(i+1,j)} \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} W_m^{(i+1,j)}$$

and for all $t \in \{1 \cdots m\}$:

$$W_t^{(i+1,j)} \triangleq \begin{cases} wx_t^{(i+1,j)} \xrightarrow{\text{po}} wbx_t^{(i+1,j)} & \text{if } q_j^{i+1}{=}p_j^i{+}1 \text{ and } \neg\texttt{committed}(w_j^{i+1}, \xi_j^{p^i_j}) \\ \emptyset & \text{otherwise} \end{cases}$$

such that $\texttt{lab}(wx_t^{(i+1,j)}) = (\texttt{W}, x_t, w_j^{i+1}[x_t], -)$ and $\texttt{lab}(wbx_t^{(i+1,j)}) = (\texttt{WB}, x_t)$.

We write T^i_{rec} for the set of all transactions recovered in the i^{th} era:

$$T^i_{rec} \triangleq \left\{ \xi \;\middle|\; \exists j.\ \texttt{lab}(rlog_{(i,j)}) = (\texttt{R}, \texttt{log}[\tau_j], \xi, -) \wedge W(i,j) \neq \emptyset \right\}$$

Let $RS^0_\xi = WS^0_\xi = \emptyset$. When ξ is a transaction of thread τ with body $\texttt{T}$, then the trace $\theta_i(\xi)$ is of the form:

$$Fs \xrightarrow{\text{po}} Ts \xrightarrow{\text{po}} dsb_1 \xrightarrow{\text{po}} log \xrightarrow{\text{po}} logwb \xrightarrow{\text{po}} PLs \xrightarrow{\text{po}} Ws \xrightarrow{\text{po}} dsb_2 \xrightarrow{\text{po}} WUs \xrightarrow{\text{po}} RUs$$

where $\texttt{lab}(dsb_1) = \texttt{lab}(dsb_2) = (\texttt{DSB},)$, and :

- Fs denotes the sequence of events failing to obtain the necessary locks, i.e. those iterations that do not succeed in promoting the writer locks;
- Ts denotes the sequence of events corresponding to the execution of $(\!|\texttt{T}|\!)$ and is of the form $t_1 \xrightarrow{\text{po}} \cdots \xrightarrow{\text{po}} t_k$, where for $m \in \{1 \cdots k\}$ each t_m is either of the form $rd(x_m, v_m, RS_{m-1}, WS_{m-1})$ or $wr(x_m, v_m, RS_{m-1}, WS_{m-1})$, with:

$$rd(x_m, v_m, RS_{m-1}, WS_{m-1}) \triangleq \begin{cases} frl_m & \text{if } x_m \notin RS_{m-1} \cup WS_{m-1} \\ \xrightarrow{\text{po}} rl^0_{x_m} \xrightarrow{\text{po}} rl_{x_m} & \\ \xrightarrow{\text{po}} wlog_{x_m} \xrightarrow{\text{po}} wrs_{x_m} & \\ \xrightarrow{\text{po}} r_{x_m} & \\ \\ wrs_{x_m} \xrightarrow{\text{po}} r_{x_m} & \text{otherwise} \end{cases}$$

$$wr(x_m, v_m, RS_{m-1}, WS_{m-1}) \triangleq \begin{cases} fs_m & \text{if } x_m \notin RS_{m-1} \cup WS_{m-1} \\ \overset{\text{po}}{\rightarrow} rl^0_{x_m} \overset{\text{po}}{\rightarrow} rl_{x_m} & \\ \overset{\text{po}}{\rightarrow} wlog_{x_m} \overset{\text{po}}{\rightarrow} wws_{x_m} & \\ \overset{\text{po}}{\rightarrow} lw_{x_m} \overset{\text{po}}{\rightarrow} lwb_{x_m} & \\ \\ wws_{x_m} \overset{\text{po}}{\rightarrow} lw_{x_m} \overset{\text{po}}{\rightarrow} lwb_{x_m} & \text{otherwise} \end{cases}$$

where frl_m denotes the sequence of events attempting (but failing) to acquire the read lock on x_m, $\mathtt{lab}(rl^0_{x_m}) = (\mathtt{R}, xl_m, a, \mathtt{Q})$, for some even value a, $\mathtt{lab}(rl_{x_m}) = (\mathtt{U}, xl_m, a, \mathtt{Q}, a+2, \mathtt{L})$, $\mathtt{lab}(wlog_{x_m}) = (\mathtt{W}, \mathtt{l[}x_m\mathtt{]}, \xi, -)$, $\mathtt{lab}(wrs_{x_m}) = (\mathtt{W}, \mathtt{RS}, RS_m, -)$, $\mathtt{lab}(r_{x_m}) = (\mathtt{R}, x_m, v_m, -)$ if $x_m \notin WS_{m-1}$; and $\mathtt{lab}(r_{x_m}) = (\mathtt{R}, \mathtt{w[}x_m\mathtt{]}, v_m, -)$ otherwise, $\mathtt{lab}(wws_{x_m}) = (\mathtt{W}, \mathtt{WS}, WS_m, -)$, $\mathtt{lab}(lw_{x_m}) = (\mathtt{W}, \mathtt{w[}x_m\mathtt{]}, v_m, -)$, $\mathtt{lab}(lwb_{x_m}) = (\mathtt{WB}, \mathtt{w[}x_m\mathtt{]})$, and for all $m > 0$:

$$RS_{m+1} \triangleq \begin{cases} RS_m \cup \{x_m\} & \text{if } t_m{=}rd(x_m, v_m, -, -) \\ RS_m & \text{otherwise} \end{cases}$$

$$WS_{m+1} \triangleq \begin{cases} WS_m \cup \{x_m\} & \text{if } t_m{=}wr(x_m, v_m, -, -) \\ WS_m & \text{otherwise} \end{cases}$$

Let $\mathtt{RS}_\xi = RS_m$ and $\mathtt{WS}_\xi = WS_m$; let $\mathtt{RS}_\xi \cup \mathtt{WS}_\xi$ be enumerated as $\{x_1 \cdots x_i\}$ for some i.

- $\mathtt{lab}(log) = (\mathtt{W}, \mathtt{ws}[\xi], \mathtt{w}, -)$, and $\mathtt{lab}(logwb) = (\mathtt{WB}, \mathtt{ws}[\xi])$.
- PLs denotes the sequence of events promoting the reader locks to writer ones (when the given location is in the write set), and is of the form $PL_{x_1} \overset{\text{po}}{\rightarrow} \cdots \overset{\text{po}}{\rightarrow} PL_{x_i}$, where for all $n \in \{1 \cdots i\}$:

$$PL_{x_n} = \begin{cases} plw_{x_n} \overset{\text{po}}{\rightarrow} spl_{x_n} \overset{\text{po}}{\rightarrow} pl_{x_n} & \text{if } x_n \in \mathtt{WS}_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

and $\mathtt{lab}(plw_{x_i}) = (\mathtt{U}, xl_i, v_i, \mathtt{Q}, v_i{-}1, \mathtt{L})$ for some even value v_i; pls_{x_i} denotes the sequence of reads waiting for the lock to be available (spinning), and $\mathtt{lab}(pl_{x_i}) = (\mathtt{R}, xl_i, 1, \mathtt{Q})$:

- Ws denotes the sequence of events committing the writes of $(\!|\mathtt{T}|\!)$ and is of the form $c_{x_1} \overset{\text{po}}{\rightarrow} \cdots \overset{\text{po}}{\rightarrow} c_{x_i}$, where for all $n \in \{1 \cdots i\}$:

$$c_{x_n} = \begin{cases} lr_{x_n} \overset{\text{po}}{\rightarrow} w_{x_n} \overset{\text{po}}{\rightarrow} wb_{x_n} & \text{if } x_n \in \mathtt{WS}_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

and $\mathtt{lab}(lr_{x_n}) = (\mathtt{R}, \mathtt{w[}x_n\mathtt{]}, v_n, -)$, $\mathtt{lab}(w_{x_n}) = (\mathtt{W}, x_n, v_n, -)$, $\mathtt{lab}(wb_{x_n}) = (\mathtt{WB}, x_n)$, for some v_n.

- WUs denotes the sequence of events releasing the writer locks and is of the form $WU_{x_1} \overset{\text{po}}{\rightarrow} \cdots \overset{\text{po}}{\rightarrow} WU_{x_i}$, where for all $n \in \{1 \cdots i\}$:

$$WU_{x_n} = \begin{cases} wu_{x_n} & \text{if } x_n \in \mathtt{WS}_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

where $\mathtt{lab}(wu_{x_n}) = (\mathtt{W}, xl_n, 0, \mathtt{L})$.

- RUs denotes the sequence of events releasing the reader locks (when the given location is in the read set only) and is of the form $RU_{x_1} \overset{\text{po}}{\rightarrow} \cdots \overset{\text{po}}{\rightarrow} RU_{x_i}$, where for all $n \in \{1 \cdots i\}$:

$$RU_{x_n} = \begin{cases} ru_{x_n} & \text{if } x_n \notin \mathtt{WS}_\xi \\ \emptyset & \text{otherwise} \end{cases}$$

where $\mathtt{lab}(ru_{x_n}) = (\mathtt{U}, xl_n, v_n, \mathtt{Q}, v_n{-}2, \mathtt{L})$ for some v_n.

Note that for all $\xi_1, \xi_2 \in T^i_{rec}$, if $\xi_1 \neq \xi_2$, then $\mathtt{WS}_{\xi_1} \cap \mathtt{WS}_{\xi_2} = \emptyset$. As such, for each location x, there is at most one write to x during the execution of the recovery $\theta_{\mathrm{init}(i)}$. We denote this write by rec_x.

For each location $x \in \mathtt{WS}_\xi$, let fw_x denote the maximal write (in po order) logging a write for x in $\mathtt{w}[x\ ]$. That is, when $Ts = t_1 \xrightarrow{\mathrm{po}} \cdots \xrightarrow{\mathrm{po}} t_m$, let $fw_x = \mathtt{wmax}(x, [t_1 \cdots t_m])$, where:

$$\mathtt{wmax}(x, [\,]) \text{ undefined}$$
$$\mathtt{wmax}(x, L.[t]) \triangleq \begin{cases} t.lw_x & \text{if } t{=}wr(x,-,-,-) \\ \mathtt{wmax}(x, L) & \text{otherwise} \end{cases}$$

Note that if an execution is PARMv8-consistent, then $(fw_{x_n}, lr_{x_n}) \in \mathrm{rf}$, for all $x_n \in \mathtt{WS}_\xi$.

B.3 Implementation Soundness

In order to establish the soundness of our implementation, it suffices to show that given an PARMv8-consistent execution graph G of the implementation, we can construct a corresponding PSER-consistent execution graph G' with the same outcome. In era i, given a transaction ξ of thread τ_j with code $\mathtt{T}$, $\mathtt{RS}_\xi \cup \mathtt{WS}_\xi = \{x_1 \cdots x_i\}$ and trace $\theta_i(\xi)$ as above with $\theta_i(\xi).Ts = t_1 \xrightarrow{\mathrm{po}} \cdots \xrightarrow{\mathrm{po}} t_k$, we construct the corresponding PSER execution trace $\theta'_i(\xi)$ as follows:

$$\theta'_i(\xi) \triangleq t'_1 \xrightarrow{\mathrm{po}} \cdots \xrightarrow{\mathrm{po}} t'_k$$

where for all $m \in \{1 \cdots k\}$:

$$\begin{array}{lll} \mathtt{lab}(t'_m){=}(\mathtt{R}, x_m, v_m, \xi) & \text{when} & t_m = rd(x_m, v_m, -, -) \\ \mathtt{lab}(t'_m){=}(\mathtt{W}, x_m, v_m, \xi) & \text{when} & t_m = wr(x_m, v_m, -, -) \end{array}$$

and in the first case the identifier of t'_m is that of $\theta_i(\xi).r_{x_m}$; and in the second case the identifier of t'_m is that of $\theta_i(\xi).lw_{x_m}$. We thus define a function, $\mathtt{imp}(.)$, mapping each PSER event t'_m to its corresponding PARMv8 event: $\theta_i(\xi).r_{x_m}$ when $\mathtt{lab}(t'_m){=}(\mathtt{R}, x_m, v_m, \xi)$, or $\theta_i(\xi).lw_{x_m}$ when $\mathtt{lab}(t'_m){=}(\mathtt{W}, x_m, v_m, \xi)$.

We are now in a position to demonstrate the soundness of our implementation. Given an PARMv8-consistent execution graph G_i of the implementation in the i^{th} era, we construct a PSER execution graph G'_i as follows and demonstrate that it is PSER-consistent:

- $G'_i.E{=}G'_i.I \cup Rec \cup Run$, with $Rec \triangleq \bigcup_{\xi \in T^i_{rec}} \theta'_{i-1}(\xi).E$, $\theta'_0(-){=}\emptyset$ and $Run \triangleq \bigcup_{\xi \in T^i} \theta'_i(\xi).E$.
- $G'_i.I = \left\{ (\mathtt{W}, x, v, 0) \;\middle|\; \begin{array}{l} x \in \mathrm{Loc} \wedge (i = 0 \Rightarrow v = 0) \wedge \\ (i > 0 \Rightarrow \exists e \in \max\left(\mathrm{nvo}_i|_{G'_{i-1}.P \cap W_x}\right).\ \mathtt{val_w}(e){=}v; \end{array} \right\}$
- $G'_i.P = G'_i.I \cup PRec \cup \bigcup_{\xi \in T^i} \mathtt{p}(\xi)$, where:

$$PRec \triangleq \begin{cases} Rec & \theta^P_{\mathrm{init}_i}{=}\theta_{\mathrm{init}_i} \wedge \theta_{\mathrm{init}_i}.E \cap D \subseteq G_i.P \\ \emptyset & \text{otherwise} \end{cases}$$

$$\mathtt{p}(\xi) \triangleq \begin{cases} \theta'_i(\xi).E & \text{if } \theta_i(\xi).E \cap D \subseteq G_i.P \\ \emptyset & \text{otherwise} \end{cases}$$

- $G'_i.\mathrm{po} = G'_i.I \times (G'_i.E \setminus G'_i.I) \cup (Rec \times Run)_i \cup G.\mathrm{po}|_{G'.E}$
- $G'_i.\mathrm{rf} = \bigcup_{\xi \in T^i} \mathtt{RF}_\xi \cup \bigcup_{\xi \in T^i_{rec}} \mathtt{RF'}_\xi$

- $G'_i.\mathsf{mo} = \left(G'_i.I \times ((G'_i.E \setminus G'_i.I) \cap W)\right)_{loc}$
  $\cup\ ((Rec \cap W) \times (Run \cap W))_{loc}$
  $\cup\ \left\{(e,e') \;\middle|\; \exists x.\ e,e' \in W_x \cap Rec \wedge \mathtt{tx}(e){=}\mathtt{tx}(e') \wedge (e,e') \in G'_i.\mathrm{po}\right\}$
  $\cup\ \mathsf{MO}$
- $G'_i.\mathsf{nvo} = G'_i.I \times ((G'_i.E \setminus G'_i.I) \cap D)$
  $\cup\ \left\{(e,e') \;\middle|\; e,e' \in G'_i.I \cap D \wedge \mathtt{id}(e) < \mathtt{id}(e')\right\}$
  $\cup\ ((Rec \cap D) \times (Run \cap D))$
  $\cup\ \left\{(e,e') \;\middle|\; e,e' \in G'_i.D \cap Rec \wedge (e,e') \in G'_i.\mathsf{st} \cap \mathrm{po}\right\}$
  $\cup\ \left\{(e,e') \;\middle|\; e,e' \in G'_i.Rec \cap D \wedge (e,e') \notin G'_i.\mathsf{st} \wedge (e,e') \in G'_i.\mathsf{hb}\right\}$
  $\cup\ \left\{(e,e') \;\middle|\; e,e' \in G'_i.Rec \cap D \wedge (e,e') \notin G'_i.\mathsf{st} \cup \mathsf{hb} \wedge \mathtt{tx}(e) \prec \mathtt{tx}(e')\right\}$
  $\cup\ \mathsf{NVO}$

where $\prec$ denotes a strict total order on transaction identifiers (e.g. natural number ordering), and:

$$\mathsf{RF}_\xi \triangleq \left\{(t'_k,t'_j) \;\middle|\; \begin{array}{l}\exists x,v,\xi.\ \mathtt{lab}(t'_j){=}(\mathtt{R},x,v,\xi) \wedge \mathtt{lab}(t'_k){=}(\mathtt{W},x,v,\xi)\\ \wedge (t_k.wl_x, t_j.r_x) \in G.\mathsf{rf}\end{array}\right\}$$
$$\cup \left\{(t'_k,t'_j) \;\middle|\; \begin{array}{l}\exists x,v,\xi,\xi'.\ \mathtt{lab}(t'_j){=}(\mathtt{R},x,v,\xi) \wedge \mathtt{lab}(t'_k){=}(\mathtt{W},x,v,\xi') \wedge \xi \neq \xi'\\ \wedge\ t_k = \theta_i(\xi').fw_x \wedge (\theta_i(\xi').w_x, \theta_i(\xi).t_j.r_x) \in G.\mathsf{rf}\end{array}\right\}$$

$$\mathsf{RF'}_\xi \triangleq \left\{(w,r) \;\middle|\; \mathtt{tx}(r){=}\xi \wedge (w,r) \in G'_{i-1}.\mathsf{rf} \wedge \mathtt{tx}(w){=}\mathtt{tx}(r)\right\}$$
$$\cup \left\{(w_0,r) \;\middle|\; \begin{array}{l}\mathtt{tx}(r){=}\xi \wedge \mathtt{loc}(r){=}\mathtt{loc}(w_0) \wedge w_0 \in G'_i.I\\ \wedge \exists w.\ (w,r) \in G'_{i-1}.\mathsf{rf} \wedge \mathtt{tx}(w) \neq \mathtt{tx}(r)\end{array}\right\}$$

$$\mathsf{MO} \triangleq \left\{(t'_k,t'_j) \;\middle|\; \mathtt{tx}(t'_k) = \mathtt{tx}(t'_j) \wedge \mathtt{loc}(t'_k){=}\mathtt{loc}(t'_j) \wedge t'_k,t'_j \in W \wedge (t_k,t_j) \in G.\mathrm{po}\right\}$$
$$\cup \left\{(t'_k,t'_j) \;\middle|\; \begin{array}{l}t'_k,t'_j \in W \wedge \exists x,\xi_k,\xi_j.\ \mathtt{loc}(t'_k){=}\mathtt{loc}(t'_j){=}x\\ \wedge\ t_k \in \theta_i(\xi_k) \wedge t_j \in \theta_i(\xi_j) \wedge (\theta_i(\xi_k).c_x, \theta_i(\xi_j).c_x) \in G.\mathsf{mo}\end{array}\right\}$$

$$\mathsf{NVO} \triangleq \left\{(t'_k,t'_j) \;\middle|\; \mathtt{tx}(t'_k) = \mathtt{tx}(t'_j) \wedge t'_k,t'_j \in D \wedge (t_k,t_j) \in G.\mathrm{po}\right\}$$
$$\cup \left\{(t'_k,t'_j) \;\middle|\; \begin{array}{l}t'_k,t'_j \in W \wedge \exists x,y,\xi_k,\xi_j.\ \mathtt{loc}(t'_k){=}x \wedge \mathtt{loc}(t'_j){=}y\\ \wedge\ t_k \in \theta_i(\xi_k) \wedge t_j \in \theta_i(\xi_j) \wedge (\theta_i(\xi_k).c_x, \theta_i(\xi_j).c_y) \in G.\mathsf{nvo}\end{array}\right\}$$

Lemma 4. *Given an PARMv8-consistent execution graph G of the implementation and its corresponding PSER execution graph G' constructed as above, for all a, b, ξ_a, ξ_b, x:*

$$\xi_a \neq \xi_b \wedge \xi_a \neq 0 \wedge \xi_a \notin T_{rec} \wedge a \in \theta'(\xi_a) \wedge b \in \theta'(\xi_b) \wedge \mathtt{loc}(a) = \mathtt{loc}(b) = x \Rightarrow$$
$$((a,b) \in G'.\mathsf{rf} \Rightarrow \theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x) \tag{1}$$
$$\wedge\ ((a,b) \in G'.\mathsf{mo} \Rightarrow \theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x) \tag{2}$$
$$\wedge\ \big((a,b) \in G'.\mathsf{rb} \Rightarrow (x \in \mathtt{WS}_{\xi_a} \wedge \theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x) \tag{3}$$
$$\vee\ (x \notin \mathtt{WS}_{\xi_a} \wedge \theta(\xi_a).ru_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x)\big)$$

Proof. Pick an arbitrary PARMv8-consistent execution graph G of the implementation and its corresponding PSER execution graph G' constructed as above. Pick an arbitrary a, b, ξ_a, ξ_b, x such that $\xi_a \neq \xi_b$, $\xi_a \neq 0$, $\xi_a \notin T_{rec}$, $a \in \theta'(\xi_a)$, $b \in \theta'(\xi_b)$, and $\mathtt{loc}(a) = \mathtt{loc}(b) = x$.

RTS. (1)
Assume $(a,b) \in G'.\mathsf{rf}$. Since $\xi_a \neq 0$, we know that $\xi_b \notin T_{rec}$. As such, from the definition of $G'.\mathsf{rf}$ we then know $(\theta(\xi_a).w_x, \theta(\xi_b).r_x) \in G.\mathsf{rf}$. On the other hand, from **??** we know that either i) $x \in \mathtt{WS}_{\xi_b}$ and $\xi_b.wu_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.rl_x$; or ii) $x \notin \mathtt{WS}_{\xi_b}$ and $\xi_b.ru_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.pl_x$; or iii) $\xi_a.wu_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_b.rl_x$.

In case (i) we then have $\xi_a.w_x \overset{G.\mathsf{rf}}{\rightarrow} \xi_b.r_x \overset{G.\mathsf{po}}{\rightarrow} \xi_b.wu_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.rl_x \overset{G.\mathsf{po}}{\rightarrow} \xi_a.w_x$. From the PARMv8-consistency of the execution we have $G.\mathsf{rf} = G.\mathsf{rf}_i \cup G.\mathsf{rf}_e \subseteq G.\mathsf{po} \cup G.\mathsf{ob}$. We thus have $\xi_a.w_x \overset{G.\mathsf{po}\cup\mathsf{ob}}{\rightarrow} \xi_b.r_x \overset{G.\mathsf{po}}{\rightarrow} \xi_b.wu_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.rl_x \overset{G.\mathsf{po}}{\rightarrow} \xi_a.w_x$. As such, since $\xi_b.wu_x$ is a release (L) write and $\xi_a.rl_x$ is an acquire (Q) read, we have $\xi_a.w_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_b.wu_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.rl_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.w_x$. That is, we have $\xi_a.w_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.w_x$, contradicting the assumption that G is PARMv8-consistent.

Similarly in case (ii) we have $\xi_a.w_x \overset{G.\mathsf{rf}}{\rightarrow} \xi_b.r_x \overset{G.\mathsf{po}}{\rightarrow} \xi_b.ru_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.pl_x \overset{G.\mathsf{po}}{\rightarrow} \xi_a.w_x$. With analogous reasoning steps we then get $\xi_a.w_x \overset{G.\mathsf{ob}}{\rightarrow} \xi_a.w_x$, contradicting the assumption that G is PARMv8-consistent.

In case (iii) the desired result holds immediately.

RTS. (2) and (3)
The proofs of these parts are analogous and are omitted here. □

Lemma 5. *Given an PARMv8-consistent execution graph G of the implementation and its corresponding PSER execution graph G' constructed as above, for all a, b:*

$$(a, b) \in G'.\mathsf{hb} \wedge a \notin G'.I \cup Rec \Rightarrow (\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{ob}$$

Proof. Let $G'.\mathsf{hb}^1 \triangleq G'.\mathsf{po}_\mathsf{T} \cup \mathsf{rf}_\mathsf{T} \cup \mathsf{mo}_\mathsf{T} \cup \mathsf{rb}_\mathsf{T}$, and $G'.\mathsf{hb}^{n+1} \triangleq G'.\mathsf{hb}^1; G'.\mathsf{hb}^n$, for all $n > 1$. We then show the following equivalent result:

$$\forall n \in \mathbb{N}^+.\ (a, b) \in G'.\mathsf{hb}^n \wedge a \notin G'.I \cup Rec \Rightarrow (\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{ob}$$

We proceed by induction on n.
Base case $n = 1$
Pick arbitrary a, b such that $(a, b) \in G'.\mathsf{hb}^1$ and $a \notin G'.I \cup Rec$. Given the definition of hb^1, we thus know that either: i) $(a, b) \in G'.\mathsf{po}_\mathsf{T}$; or ii) $(a, b) \in G'.\mathsf{rf}_\mathsf{T}$; or iii) $(a, b) \in G'.\mathsf{mo}_\mathsf{T}$; or iv) $(a, b) \in G'.\mathsf{rb}_\mathsf{T}$. In case (i), from the construction of G' we know there exists $dsb \in DSB_{\mathtt{Full}}$ such that $\mathtt{imp}(a) \overset{G.\mathsf{po}}{\rightarrow} dsb \overset{G.\mathsf{po}}{\rightarrow} \mathtt{imp}(b)$. As such, from the PARMv8-consistency of G we have $(\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{ob}$.

In case (ii), we know there exists ξ_a, ξ_b such that $\xi_a \neq \xi_b$, $\xi_a \neq 0$, $\xi_a \notin T_{rec}$, $a \in \theta'(\xi_a)$ and $b \in \theta'(\xi_b)$. As such, from Lemma 4 we have $\theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x$. We thus have $\mathtt{imp}(a) \overset{G.\mathsf{po}}{\rightarrow} \theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x \overset{G.\mathsf{po}}{\rightarrow} \mathtt{imp}(b)$. As $\xi_a.wu_x$ is a release (L) write and $\xi_b.rl_x$ is an acquire (Q) read, we have $\mathtt{imp}(a) \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x \overset{G.\mathsf{ob}}{\rightarrow} \mathtt{imp}(b)$. That is, we have $(\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{ob}$.

The proof of cases (iii-iv) cases are analogous and are omitted here.

Inductive case $n = m{+}1$ **for** $m > 0$
Pick arbitrary a, b such that $(a, b) \in G'.\mathsf{hb}^n$ and $a \notin G'.I \cup Rec$. That is, there exists c, ξ_c such that $(a, c) \in G'.\mathsf{hb}^1$, $(c, b) \in G'.\mathsf{hb}^m$ and $c \in \theta'(\xi_c)$. From the proof of the base case we then have $(\mathtt{imp}(a), \mathtt{imp}(c)) \in G.\mathsf{ob}$. Moreover, given the construction of G' and since $\xi_a \neq 0$, and $\xi_a \notin T_{rec}$, we know that $\xi_c \neq 0$, and $\xi_c \notin T_{rec}$. As such, from the inductive hypothesis we have $(\mathtt{imp}(c), \mathtt{imp}(b)) \in G.\mathsf{ob}$. As $(\mathtt{imp}(a), \mathtt{imp}(c)) \in G.\mathsf{ob}$ and $(\mathtt{imp}(c), \mathtt{imp}(b)) \in G.\mathsf{ob}$, we thus have $(\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{ob}$, as required.

□

Lemma 6 (Implementation soundness). *For all PARMv8-consistent execution graphs G of the implementation and their counterpart PSER execution graphs G' constructed as above:*

$$G'.\mathsf{hb} \text{ is irreflexive} \tag{4}$$

$$G'.\mathsf{hb} \cap (D \times D) \subseteq G'.\mathsf{nvo} \tag{5}$$

$$dom(G'.[D]; \mathsf{st}; [P]) \subseteq G'.P \tag{6}$$

Proof. Pick an arbitrary PARMv8-consistent execution G of the implementation and its counterpart PSER execution graphs G' constructed as above.

RTS. (4)
We proceed by contradiction. Let assume that there exists a such that $(a, a) \in G'.\mathsf{hb}$. Note that given the construction of G', we know that the initialisation events in $G'.I$ have no incoming $G'.\mathsf{po} \cup \mathsf{rf} \cup \mathsf{mo} \cup \mathsf{rb}$ edges, and as such this cycle contains *no initialisation events in* $G'.I$; in particular, $a \notin G'.I$ and thus $\mathtt{tx}(()a) \neq 0$. Moreover, since the only incoming $G'.\mathsf{po} \cup \mathsf{rf} \cup \mathsf{mo} \cup \mathsf{rb}$ edges to the events in $G'.Rec$ are those from the initialisation events in $G'.I$, and since this cycle contains no initialisation events, we also know that this cycle contains no events from $G'.Rec$. That is, $a \notin G'.Rec$. As such, from Lemma 5 we have $(\mathtt{imp}(a), \mathtt{imp}(a)) \in G.\mathsf{ob}$, contradicting our assumption that G is PARMv8-consistent.

RTS. (5)
Pick an arbitrary a, b such that $(a, b) \in G'.\mathsf{hb}$ and $a, b \in G'.D$; that is, $a, b \in W$. Let $\mathtt{loc}(a) = x$ and $\mathtt{loc}(b) = y$. There are now three cases to consider: i) $a \in G'.I$; or ii) $a \in G'.Rec$; or iii) $a \in G'.Run$.

In case (i), given the construction of G', we know that the initialisation events in $G'.I$ have no incoming $G'.\mathsf{po} \cup \mathsf{rf} \cup \mathsf{mo} \cup \mathsf{rb}$ edges, and thus we know that $b \notin G'.I$. Consequently, from the construction of G' we have $(a, b) \in G'.\mathsf{nvo}$.

In case (ii), given the construction of G', we know that the only outgoing $G'.\mathsf{po} \cup \mathsf{rf} \cup \mathsf{mo} \cup \mathsf{rb}$ edges of events in Rec is to events in $Rec \cup Run$. As such, we know that $b \in G'.Rec \cup Run$. Consequently, from the construction of G' we have $(a, b) \in G'.\mathsf{nvo}$.

In case (iii), given the construction of G', we know that the only outgoing $G'.\mathsf{po} \cup \mathsf{rf} \cup \mathsf{mo} \cup \mathsf{rb}$ edges of events in Run is to events in Run. As such, we know that $b \in G'.Run$. It is then straightforward to demonstrate from part (4) that $\mathtt{tx}(a) \neq \mathtt{tx}(b)$. That is, there exists ξ_a, ξ_b such that $\xi_a \neq \xi_b$, $a \in \theta'(\xi_a)$ and $b \in \theta'(\xi_b)$. There are now four cases to consider: a) $(a, b) \in G'.\mathsf{po}$; or b) $(a, b) \in G'.\mathsf{rf}$; or c) $(a, b) \in G'.\mathsf{mo}$; or d) $(a, b) \in G'.\mathsf{rb}$.

In case (a) we know there exist $dsb \in DSB_{\mathtt{full}}$, $wb \in WB$ such that $\mathtt{loc}(wb) = \mathtt{loc}(\mathtt{imp}(a))$, and $\mathtt{imp}(a) \overset{G.\mathsf{po}}{\rightarrow} wb \overset{G.\mathsf{po}}{\rightarrow} dsb \overset{G.\mathsf{po}}{\rightarrow} \mathtt{imp}(b)$; thus from the PARMv8-consistency of G we have: $(\mathtt{imp}(a), \mathtt{imp}(b)) \in G.\mathsf{nvo}$. Consequently, from the definition of G' we have $(a, b) \in G'.\mathsf{nvo}$.

In case (b) from Lemma 4 we have $\theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x$. Moreover, we know there exist $dsb \in DSB_{\mathtt{full}}$, $wb \in WB$ such that $\mathtt{loc}(wb) = \mathtt{loc}(\mathtt{imp}(a))$, and $\mathtt{imp}(a) \overset{G.\mathsf{po}}{\rightarrow} wb \overset{G.\mathsf{po}}{\rightarrow} dsb \overset{G.\mathsf{po}}{\rightarrow} \theta(\xi_a).wu_x$. As such, from the PARMv8-consistency of G we have: $(\mathtt{imp}(a), \theta(\xi_a).wu_x) \in G.\mathsf{nvo}$. Moreover, from the PARMv8-consistency of G and since $\theta(\xi_a).wu_x \overset{G.\mathsf{ob}}{\rightarrow} \theta(\xi_b).rl_x$, we have $\theta(\xi_a).wu_x \overset{G.\mathsf{mo}}{\rightarrow} \theta(\xi_b).rl_x$ and thus $\theta(\xi_a).wu_x \overset{G.\mathsf{nvo}}{\rightarrow} \theta(\xi_b).rl_x$. As such, we have $(\mathtt{imp}(a), \theta(\xi_a).wu_x) \in G.\mathsf{nvo}$. Consequently, from the definition of G' we have $(a, b) \in G'.\mathsf{nvo}$.

Proof of cases (c-d) are analogous and are omitted here.

RTS. (6) Follows immediately from the construction of G'. □