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 \(G_1 = (E_1, I_1, P, po_1, rf_1, mo_1, nvo_1)\) and \(G_2 = (E_2, I_2, P, po_2, rf_2, mo_2, nvo_2)\), whenever \(G_1\) and \(G_2\) contain disjoint events \((E_1 \cap E_2 = \emptyset)\), then their composition, written \(\text{flat}(G_1, G_2)\), is given by \((E, I, P, po, rf, mo, nvo)\), where \(E \triangleq E'_1 \cup E'_2\) with \(E'_1 \triangleq \left\{ e \in G_1.T \mid |e|_{\text{at}} \cap D \subseteq P_1 \right\}\) and \(E'_2 \triangleq E_2 \setminus I_2\), \(P \triangleq (E'_1 \cap P_1) \cup (E'_2 \cap P_2)\), \(I \triangleq I_1\), \(po \triangleq po_1|E'_1 \cup po_2|E'_2 \cup (E'_1 \times E'_2)\cup (I_1 \times I'_2)\), and:

\[
rf \triangleq rf_1|E'_1 \cup rf_2|E'_2 \cup \{(w, r) \mid \exists w' \in I_2. (w', r) \in rf_2 \wedge w = \max(nvo_1|p_1 \cup w_{\text{loc}(r)})\}
\]

\[
mo \triangleq mo_1|E'_1 \cup mo_2|E'_2 \cup (E'_1 \cap W \cap W_{\text{loc}}) \cup I_1 \times E_2 \cup W_{\text{loc}}
\]

\[
nvo \triangleq nvo_1|E'_1 \cup nvo_2|E'_2 \cup (E'_1 \cap D \cap E'_2) \cup (E'_1 \cap D \cap W) \cup I_1 \times E_2 \cap D
\]

Given an execution chain \(C = G_1, \ldots, G_n\), its flattened execution, written \(\text{flat}(C)\), is given by \(G'_n\), where \(G'_1 \triangleq G_1\) and \(G'_i \triangleq \text{flat}(G'_{i-1}, G_i)\) for \(i \in \{2, \ldots, n\}\).

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

1. \((\langle - , G.E, G.po \rangle) \in \text{interp}(P)\); and
2. if \(G_2 \in \text{exec}(\text{rec}_{\text{PSER}}(P, G_1))\), then there exists \(\nu \in \text{Val} \) such that \(\langle \nu, G.E, G.po \rangle \in \text{interp}(P)\).
Proof. Pick arbitrary $P, G_1 \in \text{pexec}(P), G_2 \in \text{pexec}(\text{rec}_{\text{PSER}}(P, G_1))$ such that $G_1$ and $G_2$ are PSER-consistent. Let $F = (P, \text{rec}_{\text{PSER}})$, $G = \text{flat}(G_1, G_2)$ and $F' = \text{rec}_{\text{PSER}}(P, G_1)$.

**RTS. (1)**
Let us write $\xi_i,j$ for $(\tau_i,j)$; given an execution $G'$, we write $G'.\xi$ for $G'.E \cap \{a \mid t(x(a)) = \xi\}$. For each $\tau_i \in \text{dom}(P)$, 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 $\text{dom}(P) = \{\xi_1, \ldots, \xi_n\}$, we have $G_1.P = \bigcup_{k=1}^m (G_1, \xi_{i,k} \cap D) \cup \bigcup_{k=1}^m (G_2, \xi_{n,k} \cap D).

Consequently, from the definition of $\text{rec}_{\text{PSER}}$ we know $F'(\tau_i) = \text{sub}(P(\tau_i), m_i+1)$. That is, there exists $M_i$ such that $P(\tau_i) = [C_1]; \cdots ; [C_{m_i}]; \cdots ; [C_{M_i}]$ and $P'(\tau_i) = [C_{m_i+1}]; \cdots ; [C_{M_i}]$, for some $C_1 \cdots C_{M_i}$. As such, given the definitions of $G, G_2$ and the semantics of PSER programs (Fig. 5), we have $\langle \cdot, \langle G.E, G.po \rangle \rangle \subseteq \text{pexec}(P)$, as required.

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

Lemma 2. For all $F = (P, \text{rec}_{\text{PSER}}), G_1 \in \text{pexec}(P), G_2 \in \text{pexec}(\text{rec}_{\text{PSER}}(P, G_1))$ and $G = \text{flat}(G_1, G_2)$, if $G_1$ and $G_2$ are PSER-consistent, then:

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

Proof. Pick arbitrary $P, G_1 \in \text{pexec}(P), G_2 \in \text{pexec}(\text{rec}_{\text{PSER}}(P, G_1))$ such that $G_1$ and $G_2$ are PSER-consistent. Let $F = (P, \text{rec}_{\text{PSER}})$ and $G = \text{flat}(G_1, G_2)$.

**RTS. (1)**
From Lemma 1 we know $\langle \cdot, \langle G.E, G.po \rangle \rangle \in \{P\}_{\text{PSER}}$. On the other hand, from the definition of $\text{flat}(\cdot)$ we know that $G \in \text{Exec}$. As such, from the definition of $\text{pexec}(\cdot)$ we have $G \in \text{pexec}(P)$.

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

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

For the first part, from construction of $G$ we have $G, rf \subseteq G_1, rf \cup G_2, rf; G, mo \subseteq G_1, mo \cup G_2, mo;$ and $G, rb \subseteq G_1, rb \cup G_2, rb$. 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 $(a, a) \in G, \text{hb}_{\text{ser}}$. There are two cases to consider: 1) $\exists c \in G_1, E, d \in G_2, E. a \xrightarrow{G, \text{hb}_{\text{ser}}} c \xrightarrow{G, \text{hb}_{\text{ser}}} d \xrightarrow{G, \text{hb}_{\text{ser}}} a$; or 2) $\not\exists c \in G_1, E, d \in G_2, E. a \xrightarrow{G, \text{hb}_{\text{ser}}} c \xrightarrow{G, \text{hb}_{\text{ser}}} d \xrightarrow{G, \text{hb}_{\text{ser}}} a$. In case (1) we thus have $a \xrightarrow{G, \text{hb}_{\text{ser}}} c$ and thus $(a, c) \in G, \text{hb}_{\text{ser}} \cap (G_2, E \times G_1, E).$ This however leads to a contradiction since as described above.
we have $G.\text{hb}_{\text{ser}} \cap (G_2.E \times G_1.E) = \emptyset$. In case (2) since $G.\text{hb}_{\text{ser}} \cap (G_2.E \times G_1.E) = \emptyset$, we have $(a, a) \in G_1.\text{hb}_{\text{ser}} \cup G_2.\text{hb}_{\text{ser}}$. This however contradicts our assumption that $G_1, G_2$ are PSER-consistent.

For the third part, pick an arbitrary $(a, b) \in G.\text{hb}_{\text{ser}} \cap (D \times D)$. As $G.\text{hb}_{\text{ser}} \subseteq G_1.\text{hb}_{\text{ser}} \cup G_2.\text{hb}_{\text{ser}} \cup (G_1.E \times G_2.E)$, we know either $(a, b) \in G_1.\text{hb}_{\text{ser}} \cup G_2.\text{hb}_{\text{ser}}$, 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 $(a, b) \in G.\text{nvo}$, as required.

For the fourth part, pick an arbitrary $(a, b) \in dom([D]; G.\text{st}; [G.P])$; that is there exists $b$ such that $a \in D, (a, b) \in G.\text{st}$ 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 $(a, b) \in G_1.\text{st}$; or 2) $a, b \in G_2.E, b \in G_1.P$ and $(a, b) \in G_1.\text{st}$. 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 $(a, a) \in G.\text{nvo}^+$. There are two cases to consider: 1) $\exists c \in G_1.E, d \in G_2.E$. $a \xrightarrow{G.\text{nvo}} c \xrightarrow{G.\text{nvo}} d \xrightarrow{G.\text{nvo}} a$; or 2) $\exists c \in G_1.E, d \in G_2.E$. $a \xrightarrow{G.\text{nvo}} c \xrightarrow{G.\text{nvo}} d \xrightarrow{G.\text{nvo}} a$. In case (1) we thus have $d \xrightarrow{G.\text{nvo}} a \rightarrow c$ and thus $(d, c) \in G.\text{nvo}^+ \cap (G_2.E \times G_1.E)$. That is, there exists $d', c'$ such that $(d', c') \in G.\text{nvo}^+ \cap (G_2.E \times G_1.E)$. This however leads to a contradiction since as described above we have $G.\text{nvo} \cap (G_2.E \times G_1.E) = \emptyset$. In case (2) since $G.\text{nvo} \cap (G_2.E \times G_1.E) = \emptyset$, we have $(a, a) \in G_1.\text{nvo}^+ \cup G_2.\text{nvo}^+$; i.e. $\exists b, (b, b) \in G_1.\text{nvo} \cup G_2.\text{nvo}$. This however contradicts our assumption that $G_1, G_2$ are PSER-consistent.

**RTS. (2)**
From Lemma 1 we know there exists $\nu \in \text{Val}$ such that $(\nu, (G.E, G.\text{po})) \in \{P\}_{\text{PSER}}$. On the other hand, from the definition of $\text{fLat}(.)$ we know that $G \in \text{Exec}$. As such, from the definition of $\text{exec}(.)$ we have $G \in \text{exec}(P)$, as required.

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

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

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

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

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

**Proof.** Pick an arbitrary library $L$, a sequentially sound implementation $I$ of $L$, and program $P$.

**RTS. (1)**
Pick an arbitrary $G \in \text{exec}(\text{psr}(P, L))$ 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 $S = G.T \setminus G.E \subseteq R$
and thus $[S]; G.\text{hb}_\text{ser} \cup G.\text{nvo}_T; [G.T] = \emptyset$. It then suffices to show that there exists a sequential history of $G.T$ that is $L$-legal.

Let $A \triangleq \{ (tx(a), tx(b)) \mid (a, b) \in G.\text{hb}_\text{ser} \cup G.\text{nvo}_T \}$. From Lemma 3 and the transitivity of $G.\text{hb}_\text{ser}$ and $G.\text{nvo}_T$ we know that $A$ is a strict partial order. Let $txo$ denote a total extension of $A$ on $\{ \xi \mid \exists a \in G.E. tx(a) = \xi \}$. Let:

$$txo \triangleq \{ (a, b) \mid (a, b) \in po \cap st \lor (tx(a), tx(b)) \in txo \}$$

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

**RTS. (2)**

Pick an arbitrary $G_1, \cdots, G_n$ such that $C$ is PSER-valid. For $i \in \{ 1 \cdots n \}$, let $E_i \triangleq \{ e \in G_i.T \mid [e]_st \cap D \subseteq G_i.P \}$. 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 $G = \text{flat}(C) = G_1; \cdots; G_n$. 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 $to \triangleq to_1|_{G.E}; \cdots; to_n|_{G.E}$. From the definition of $to$ we then have:

$$to = to_1; to_2|_{G_2.E \setminus G_2.I}; \cdots; to_n|_{G_n.E \setminus G_n.I}$$

Moreover, from the definition of $to$ and each $to_i$ we then know that $to$ is a sequential history of $G$. Consequently, since $I$ is sequentially sound, we know that $to$ is $L$-legal, as required. □
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 and 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 $C = G_1; \ldots ; G_n$ of $P$ with $G_i = (E_i, I_i, P_i, p_{i1}, \ldots , p_{i, m_i}, w_{i1}, \ldots , w_{in})$, 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 $\tau_0$ 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 $\tau_1 \ldots \tau_k$, 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 (DSB$_{\text{full}}$), for each thread $\tau_j$, we know that the set of persistent events in execution era $i$, namely $P_i$, contains roughly a prefix (in po order) of thread $\tau_j$’s trace. More concretely, for each constituent thread $\tau_j \in \{\tau_1 \ldots \tau_k\} = \text{dom}(P)$, there exist $p_{i1}' \ldots \cdot p_{ik}'$, $q_{i1}' \ldots q_{ik}'$, $w_{i1}' \ldots , w_{in}'$ such that:

1. $P[\tau_j] = T_{j1}^0; \ldots ; T_{j1}^{p_{1j}'}; \ldots \cdot T_{jk}^{p_{k,j}'}; \ldots ; T_{jk}^{p_{k,j}'} = \vdots \cdot T_{jk}^{p_{k,j}'}$, where each $T_{jk}$ denotes the $k$th transaction of thread $\tau_j$; and $T_{jk}^{p_{k,j}'}$ denotes the last transaction of $\tau_j$ logged in the $i$th era, i.e. the $i$th crash occurred when $\log[\tau_j] = T_{jk}^{p_{k,j}'}$.

2. At the beginning of each execution era $i \in \{1 \ldots n\}$, for all $j$, the program executed by thread $\tau_j$ (calculated in $P$’ and subsequently executed by calling run($P$’)) is that of $\text{sub}(P[\tau_j], q_{ij}'$, such that either $q_{ij}' = p_{ij}' - 1$ when $w_{ij}' \neq \bot$, or $q_{ij}' = p_{ij}' - 1$ when $w_{ij}' = \bot$, where $p_{ij}' = 0$.

3. In each execution era $i \in \{1 \ldots n\}$, the trace of the program is of the form $\theta_{\text{init}(i)} \xrightarrow{\text{po}} \theta_{(i,j)}$, $\theta_{(i,j)}$ denotes the execution of the initialisation or recovery mechanism defined shortly, and $\theta_{(i,j)}$ denotes the trace of the $j$th constituent thread $\tau_j \in \text{dom}(P)$ and is defined as follows:

$$\theta_{(i,j)} \xrightarrow{\text{}} \begin{cases} \theta_{(i,j)}' & \text{if } \theta_{\text{init}(i)}' = \theta_{\text{init}(i)} \\ 0 & \text{otherwise} \end{cases}$$
More concretely, whenever \( \theta^{\rho}_i = \theta^{\rho}_{\text{init}} \), i.e. no crash occurred during the execution of \( \theta^{\rho}_{\text{init}} \), then \( \theta_i(\xi) \) denotes the execution of the \( q_i^{\rho} \) events releasing all locks, case all its durable events have persisted. We write \( T^i \) for the set of all transactions executed in the \( i \)th era.

Moreover, due to the placement of the \( \text{DSB}_i \) instructions, before crashing and proceeding to the next era, all durable events in \( \theta_i(\xi_i^{\rho}) \) have persisted, and a subset of the durable events in \( \theta_i(\xi_i^{\rho}) \) have persisted. Note that this subset may be equal to \( \theta_i(\xi_i^{\rho}) \), in which case all its durable events have persisted.

In the very first era \((i = 1)\) we have \( \theta^{\rho}_{\text{init}(1)} = \emptyset \), and when \( i > 1 \), the \( \theta^{\rho}_{\text{init}(i)} \) is of the form: \( \mathcal{U}s \to C(i,1) \to W(i,1) \to \cdots \to C(i,k) \to W(i,k) \to \text{dsb} \), where \( \mathcal{U}s \) denotes the sequence of events releasing all locks, \( \text{lab}(\text{dsb})=(\text{DSB, full}) \), and for all \( i \in \{1 \cdots n\} \) and \( j \in \{1 \cdots k\} \):

\[
C(i+1,j) \triangleq \text{log}_{(i+1,j)}^\rho \to \text{rwm}_{\text{p}(i+1,j)}^\rho \to \text{wp}_{(i+1,j)}^\rho
\]

where \( \text{lab}(\text{log}_{(i+1,j)}^\rho) = (R, \log[\tau_j], \xi_i^{\rho}, -) \), \( \text{lab}(\text{rwm}_{\text{p}(i+1,j)}^\rho) = (R, \text{ws}[\xi_i^{\rho}], \text{w}^{i+1}_j, -) \), \( \text{lab}(\text{wp}_{(i+1,j)}^\rho) = (W, \text{P'}[\tau_j], \text{w}^{i+1}_j, -) \), and when \( \text{dom}(\text{w}^{i+1}_j) = x_1 \cdots x_m \):

\[
W(i+1,j) \triangleq W^{(i+1,j)}_1 \to \cdots \to W^{(i+1,j)}_m
\]

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

\[
W^{(i+1,j)}_t \triangleq \begin{cases} \text{wx}^{(i+1,j)}_t^\rho \to \text{wbx}^{(i+1,j)}_t^\rho \text{ if } q_i^{j+1} = p_i^{j+1} \text{ and } \neg \text{committed}(w_j^{i+1}, \xi_i^{\rho}) \\ \emptyset \text{ otherwise} \end{cases}
\]

such that \( \text{lab}(\text{wx}^{(i+1,j)}_t^\rho) = (W, x_t, w_j^{i+1}[x_t], -) \) and \( \text{lab}(\text{wbx}^{(i+1,j)}_t^\rho) = (\text{WB}, x_t) \).

We write \( T_{\text{rec}}^i \) for the set of all transactions recovered in the \( i \)th era:

\[
T_{\text{rec}}^i \triangleq \{ \xi \mid \exists j. \text{lab}(\text{log}_{\text{p}(i,j)}^\rho) = (R, \log[\tau_j], \xi, -) \land W(i,j) \neq \emptyset \}
\]

Let \( R^{\rho}_{\xi_i} = W^{\rho}_{\xi_i} = \emptyset \). When \( \xi \) is a transaction of thread \( \tau \) with body \( T \), then the trace \( \theta_i(\xi) \) is of the form:

\[
\mathcal{F}_{\xi_i} \to \mathcal{T}_{\xi_i} \to \text{dsb}_1 \to \text{log} \to \text{logwb} \to \text{PLs} \to \mathcal{W}_{\xi} \to \text{dsb}_2 \to \text{WUs} \to \text{RUs}
\]

where \( \text{lab}(\text{dsb}_1) = \text{lab}(\text{dsb}_2) = (\text{DSB},) \), and:

- \( \mathcal{F}_{\xi} \) denotes the sequence of events failing to obtain the necessary locks, i.e. those iterations that do not succeed in promoting the writer locks;
- \( \mathcal{T}_{\xi} \) denotes the sequence of events corresponding to the execution of \( \{T\} \) and is of the form \( t_1 \to \cdots \to t_k \), where for \( m \in \{1 \cdots k\} \) each \( t_m \) is either of the form \( rd(x_m, \upsilon_m, RS_{m-1}, WS_{m-1}) \) or \( wr(x_m, \upsilon_m, RS_{m-1}, WS_{m-1}) \), with:

\[
rd(x_m, \upsilon_m, RS_{m-1}, WS_{m-1}) \triangleq \begin{cases} frl_m \to rls_m \to \text{wr}_{s_m} \to r_{s_m} \to r_{m} \text{ if } x_m \notin RS_{m-1} \cup WS_{m-1} \\ \text{wr}_{s_m} \to r_{s_m} \to r_{m} \text{ otherwise} \end{cases}
\]
where \( f \) denotes the sequence of events attempting (but failing) to acquire the read lock on \( x \), \( \text{lab}(\text{rl}_x) = (R, x.l, a, Q) \), for some even value \( a \), \( \text{lab}(\text{rl}_x) = (U, x.l, a, Q, a + 2, L) \), \( \text{lab}(\text{lwb}_x) = (W, l[l_x], \xi, -) \), \( \text{lab}(\text{wrs}_x) = (W, R, S, S_m, -) \), \( \text{lab}(\text{rl}_x) = (R, x.m, v_m, -) \) if \( x \) is not \( S_m \); and \( \text{lab}(\text{rl}_x) = (R, w[x_m], v_m, -) \) otherwise, \( \text{lab}(\text{wrs}_x) = (W, S, S_m, -) \), \( \text{lab}(\text{lwb}_x) = (W, b,w_x) \), and for all \( m > 0 \):

\[
\begin{align*}
\text{RS}_m & \triangleq \begin{cases} 
\text{RS}_m \cup \{x\} & \text{if } t = \text{rd}(x.m, v_m, -) \\
\text{RS}_m & \text{otherwise}
\end{cases} \\
\text{WS}_m & \triangleq \begin{cases} 
\text{WS}_m \cup \{x\} & \text{if } t = \text{wr}(x.m, v_m, -) \\
\text{WS}_m & \text{otherwise}
\end{cases}
\end{align*}
\]

Let \( \text{RS}_x = \text{RS}_m \) and \( \text{WS}_x = \text{WS}_m \); let \( \text{RS}_x \cup \text{WS}_x \) be enumerated as \( \{x_1 \ldots x_i\} \) for some \( i \).

- \( \text{lab}(\text{log}) = (W, w[s\xi], w, -) \), and \( \text{lab}(\text{logwb}) = (W, w[s\xi]) \).
- \( \text{PLs} \) 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 \( \text{PL}_{x_i} \rightarrow \cdots \rightarrow \text{PL}_{x_i} \), where for all \( n \in \{1 \ldots i\}:

\[
\text{PL}_{x_n} = \begin{cases} 
\text{pl}_{x_n} \rightarrow \text{sp}_{x_n} \rightarrow \text{pl}_{x_n} & \text{if } x_n \in \text{WS}_x \\
\emptyset & \text{otherwise}
\end{cases}
\]

and \( \text{lab}(\text{pl}_{x_n}) = (U, x.l, v_i, Q, v_i - 1, L) \) for some even value \( v_i \); \( \text{pls}_{x_i} \) denotes the sequence of reads waiting for the lock to be available (spinning), and \( \text{lab}(\text{pl}_{x_n}) = (R, x.l, 1, Q) \):

- \( \text{Wx} \) denotes the sequence of events committing the writes of \( \|T\| \) and is of the form \( \text{c}_{x_i} \rightarrow \cdots \rightarrow \text{c}_{x_i} \), where for all \( n \in \{1 \ldots i\}:

\[
\text{c}_{x_n} = \begin{cases} 
\text{tr}_{x_n} \rightarrow \text{w}_{x_n} \rightarrow \text{wb}_{x_n} & \text{if } x_n \in \text{WS}_x \\
\emptyset & \text{otherwise}
\end{cases}
\]

and \( \text{lab}(\text{tr}_{x_n}) = (R, w[x_n], v_n, -) \), \( \text{lab}(\text{w}_{x_n}) = (W, x_n, v_n, -) \), \( \text{lab}(\text{wb}_{x_n}) = (W, b, w_n) \), for some \( v_n \).

- \( \text{Wx} \) denotes the sequence of events releasing the writer locks and is of the form \( \text{Wx}_{x_i} \rightarrow \cdots \rightarrow \text{Wx}_{x_i} \), where for all \( n \in \{1 \ldots i\}:

\[
\text{Wx}_{x_n} = \begin{cases} 
\text{wu}_{x_n} & \text{if } x_n \in \text{WS}_x \\
\emptyset & \text{otherwise}
\end{cases}
\]

where \( \text{lab}(\text{wu}_{x_n}) = (W, x.l, 0, L) \).

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

\[
\text{Rx}_{x_n} = \begin{cases} 
\text{ru}_{x_n} & \text{if } x_n \notin \text{WS}_x \\
\emptyset & \text{otherwise}
\end{cases}
\]

where \( \text{lab}(\text{ru}_{x_n}) = (U, x.l, v_n, Q, v_n - 2, L) \) for some \( v_n \).
Note that for all $\xi_1, \xi_2 \in T_{rec}$, if $\xi_1 \neq \xi_2$, then $WS_{\xi_1} \cap WS_{\xi_2} = \emptyset$. As such, for each location $x$, there is at most one write to $x$ during the execution of the recovery $\theta_{init(i)}$. We denote this write by $rec_x$.

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

\[
\begin{align*}
    wmax(x, []) &= \text{undefined} \\
    wmax(x, L.[] t) &= \begin{cases} 
    t.\mathit{lw}_x & \text{if } t = \mathit{wr}(x, -, -, -) \\
    wmax(x, L) & \text{otherwise}
    \end{cases}
\end{align*}
\]

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

### 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 $\xi$ of thread $\tau_j$ with code $T$, $RS_\xi \cup WS_\xi = \{x_1 \cdots x_i\}$ and trace $\theta_i(\xi)$ as above with $\theta_i(\xi).Ts = t_1 \rightarrow^{po} \cdots \rightarrow^{po} t_k$, we construct the corresponding PSER execution trace $\theta'_i(\xi)$ as follows:

\[
\theta'_i(\xi) = t'_1 \rightarrow^{po} \cdots \rightarrow^{po} t'_k
\]

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

\[
\begin{align*}
    \text{lab}(t'_m) &= (R, x_m, v_m, \xi) \quad \text{when } t_m = rd(x_m, v_m, -, -) \\
    \text{lab}(t'_m) &= (W, x_m, v_m, \xi) \quad \text{when } t_m = \mathit{wr}(x_m, v_m, -, -)
    \end{align*}
\]

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 $\theta_i(\xi).w_{x_m}$. We thus define a function, $\mathit{imp}(\cdot)$, mapping each PSER event $t'_m$ to its corresponding PARMv8 event: $\theta_i(\xi).r_{x_m}$ when $\text{lab}(t'_m) = (R, x_m, v_m, \xi)$, or $\theta_i(\xi).w_{x_m}$ when $\text{lab}(t'_m) = (W, x_m, v_m, \xi)$.

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:

- $G'_i.\mathit{E} = G'_i.\mathit{I} \cup \mathit{Rec} \cup \mathit{Run}$, with $\mathit{Rec} \triangleq \bigcup_{\xi \in T_{rec}} \theta'_{i-1}(\xi).E, \theta'_i(\cdot) = \emptyset$ and $\mathit{Run} \triangleq \bigcup_{\xi \in T_i} \theta'_i(\cdot).E.$

- $G'_i.\mathit{I} = \left\{ (w, x, v, 0) \mid \begin{array}{l} x \in \text{Loc} \land (i = 0 \Rightarrow v = 0) \land \cr (i > 0 \Rightarrow \exists e \in \max(w_{\mathit{max}}) \cdot \mathit{nvo}_{\mathit{i-1}} \cap W_{\xi}) \land \mathit{val}_w(e) = v \end{array} \right\}$

- $G'_i.\mathit{P} = G'_i.\mathit{I} \cup \mathit{PRec} \cup \bigcup_{\xi \in T_i} \mathit{p}(\xi)$, where:

\[
\begin{align*}
    \mathit{PRec} & \triangleq \begin{cases} 
    \mathit{Rec} & \text{if } \theta'_1 = \theta_{init_i} \land \theta_{init_i}.E \cap D \subseteq G_i.\mathit{P} \\
    \emptyset & \text{otherwise}
    \end{cases} \\
    \mathit{p}(\xi) & \triangleq \begin{cases} 
    \theta'_i.\mathit{E} & \text{if } \theta_i.\mathit{E} \cap D \subseteq G_i.\mathit{P} \\
    \emptyset & \text{otherwise}
    \end{cases}
\end{align*}
\]

- $G'_i.\mathit{po} = G'_i.\mathit{I} \times \left( (G'_i.\mathit{E} \setminus G'_i.\mathit{I}) \cup (\mathit{Rec} \times \mathit{Run}) \cup G.\mathit{po}_{G'_i.\mathit{E}} \right)$

- $G'_i.\mathit{rf} = \bigcup_{\xi \in T_i} \mathit{RF}_\xi \cup \bigcup_{\xi \in T_{rec}} \mathit{RF}'_\xi$
\( G'_{i, mo} = \left( G'_i \times (G'_i.E \setminus G'_i.I) \cap W \right)_{loc} \)
\( \cup \left( (Rec \cap W) \times (Run \cap W) \right)_{loc} \)
\( \cup \left\{ (e, e') \mid \exists x. e, e' \in W_x \cap Rec \land tx(e) = tx(e') \land (e, e') \in G'_i.po \right\} \)
\( \cup \ MO \)
\( G'_{i, nvo} = \left( G'_i \times (G'_i.E \setminus G'_i.I) \cap D \right) \)
\( \cup \left\{ (e, e') \mid e, e' \in G'_i.D \cap Rec \land (e, e') \in G'_i.st \cap po \right\} \)
\( \cup \left\{ (e, e') \mid e, e' \in G'_i.Res \cap D \land (e, e') \notin G'_i.st \land (e, e') \in G'_i hb \right\} \)
\( \cup \left\{ (e, e') \mid e, e' \in G'_i.Res \cap D \land (e, e') \notin G'_i.st \cup \text{hb} \land tx(e) < tx(e') \right\} \)
\( \cup \ NVO \)

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

\[
RF_{\xi} \triangleq \left\{ (t'_k, t'_j) \mid \exists x, v, \xi. lab(t'_k) = (R, x, v, \xi) \land lab(t'_j) = (W, x, v, \xi) \land (t_k, w_x, t_j, r_x) \in G.rf \right\}
\]
\[
RF'_{\xi} \triangleq \left\{ (w, r) \mid tx(r) = \xi \land (w, r) \in G'_{i-1}.rf \land tx(w) = tx(r) \right\}
\]
\[
MO \triangleq \left\{ (t'_k, t'_j) \mid \exists x, \xi, t_k \in G.mo \land \exists y. (x, \xi) = x \lor t_k \in t_k(\xi) \land t_j \in t_j(\xi) \land (\theta(\xi), c_x, \theta(\xi), c_y) \in (G.rf, t'_k, t'_j) \right\}
\]
\[
NVO \triangleq \left\{ (t'_k, t'_j) \mid \exists x, \xi, t_k \in G.nvo \land \exists y. (x, \xi) = x \lor t_k \in t_k(\xi) \land t_j \in t_j(\xi) \land (\theta(\xi), c_x, \theta(\xi), c_y) \in (G.nvo, t'_k, t'_j) \right\}
\]

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

\[
\xi_a \neq \xi_b \land \xi_a \neq 0 \land \xi_a \notin T_{rec} \land a \in \theta(\xi_a) \land b \in \theta(\xi_b) \land \text{loc}(a) = \text{loc}(b) = x \Rightarrow (a, b) \in G'.rf \Rightarrow \theta(\xi_a).w_x^{G,ob} \theta(\xi_b).r_x^{G,ob} \tag{1}
\]
\[
\land ((a, b) \in G'.mo \Rightarrow \theta(\xi_a).w_x^{G,ob} \theta(\xi_b).r_x^{G,ob}) \tag{2}
\]
\[
\land ((a, b) \in G'.rb \Rightarrow (x \notin WS_{\xi_a} \land \text{loc}(a) \land \text{loc}(b) = x) \Rightarrow (x \notin WS_{\xi_a} \land \text{loc}(a) \land \text{loc}(b) = x) \Rightarrow (a, b) \in G'.rf \Rightarrow \theta(\xi_a).w_x^{G,ob} \theta(\xi_b).r_x^{G,ob}) \tag{3}
\]

**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 \( a, b, \xi_a, \xi_b, x \) 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 \( \text{loc}(a) = \text{loc}(b) = x \).

**RTS. (1)**

Assume \((a, b) \in G'.rf\). Since \( \xi_a \neq 0 \), we know that \( \xi_b \notin T_{rec} \). As such, from the definition of \( G'.rf \) we then know \((\theta(\xi_a).w_x, \theta(\xi_b).r_x) \in G.rf\). On the other hand, from ?? we know that either i) \( x \in WS_{\xi_b} \) and \( \xi_b.w_x^{G,ob} \theta(\xi_b).r_x^{G,ob} \); or ii) \( x \notin WS_{\xi_b} \) and \( \xi_b.r_x^{G,ob} \theta(\xi_b).r_x^{G,ob} \); or iii) \( \xi_a.w_x^{G,ob} \theta(\xi_b).r_x^{G,ob} \).

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

Similarly in case (ii) we have \( \xi_a.w_x \xrightarrow{G,rf} \xi_b.r_x \xrightarrow{G,po} \xi_b.r_x \xrightarrow{G,ob} \xi_a.p_l_x \xrightarrow{G,pb} \xi_a.w_x \). With analogous reasoning steps we then get \( \xi_a.w_x \xrightarrow{G,ob} \xi_a.w_x \), contradicting the assumption that \( G \) is PARMv8-consistent.

In case (iii) the desired result holds immediately.

\( \square \)

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

\[
(a, b) \in G'.hb \land a \notin G'.I \cup Rec \Rightarrow (imp(a), imp(b)) \in G.ob
\]

**Proof.** Let \( G'.hb^1 \triangleq G'.po_T \cup rf_T \cup mo_T \cup rb_T, \) and \( G'.hb^{n+1} \triangleq G'.hb^1; G'.hb^n, \) for all \( n > 1 \). We then show the following equivalent result:

\[
\forall n \in \mathbb{N}^+. (a, b) \in G'.hb^n \land a \notin G'.I \cup Rec \Rightarrow (imp(a), imp(b)) \in G.ob
\]

We proceed by induction on \( n \).

**Base case** \( n = 1 \)

Pick arbitrary \( a, b \) such that \( (a, b) \in G'.hb^1 \) and \( a \notin G'.I \cup Rec \). Given the definition of \( hb^1 \), we thus know that either: i) \( (a, b) \in G'.po_T \); or ii) \( (a, b) \in G'.rf_T \); or iii) \( (a, b) \in G'.mo_T \); or iv) \( (a, b) \in G'.rb_T \). In case (i), from the construction of \( G' \) we know there exists \( dsb \in DSB_{\text{full}} \) such that \( imp(a) \xrightarrow{G,po} dsb \xrightarrow{G,pb} imp(b) \). As such, from the PARMv8-consistency of \( G \) we have \( (imp(a), imp(b)) \in G.ob \).

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

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 \( (a, b) \in G'.hb^n \) and \( a \notin G'.I \cup Rec \). That is, there exists \( c, \xi_c \) such that \( (a, c) \in G'.hb^1 \), \( (b, c) \in G'.hb^n \) and \( c \in \theta'(\xi_c) \). From the proof of the base case we then have \( (imp(a), imp(c)) \in G.ob \). Moreover, given the construction of \( G' \) and since \( \xi_a \neq 0, \) and \( \xi_a \notin T_{\text{rec}}, \) we know that \( \xi_c = 0, \) and \( c \notin T_{\text{rec}} \). As such, from the inductive hypothesis we have \( (imp(c), imp(b)) \in G.ob \). As \( (imp(a), imp(c)) \in G.ob \) and \( (imp(c), imp(b)) \in G.ob \), we thus have \( (imp(a), imp(b)) \in G.ob \), 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:

\[
\begin{align*}
G'.\text{hb} & \text{ is irreflexive} \\ G'.\text{hb} \cap (D \times D) & \subseteq G'.\text{nvo} \\ \text{dom}(G'.[D]; \text{st}; [P]) & \subseteq G'.P
\end{align*}
\]

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 $(a, a) \in G'.\text{hb}$. Note that given the construction of $G'$, we know that the initialisation events in $G'.I$ have no incoming $G'.\text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb}$ edges, and as such this cycle contains no initialisation events in $G'.I$; in particular, $a \notin G'.I$ and thus $\text{tx}((a)) \neq 0$. Moreover, since the only incoming $G'.\text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb}$ edges to the events in $G'.\text{Rec}$ 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 $G'.\text{Rec}$. That is, $a \notin G'.\text{Rec}$. As such, from Lemma 5 we have $(\text{imp}(a), \text{imp}(a)) \in G.\text{ob}$, contradicting our assumption that $G$ is PARMv8-consistent.

RTS. (5)

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

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

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

In case (iii), given the construction of $G'$, we know that the only outgoing $G'.\text{po} \cup \text{rf} \cup \text{mo} \cup \text{rb}$ edges of events in $\text{Run}$ is to events in $\text{Run}$. As such, we know that $b \notin G'.\text{Run}$. It is then straightforward to demonstrate from part (4) that $\text{tx}(a) \neq \text{tx}(b)$. That is, there exists $\xi_a, \xi_b$ 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) $(a, b) \in G'.\text{po}$; or b) $(a, b) \in G'.\text{rf}$; or c) $(a, b) \in G'.\text{mo}$; or d) $(a, b) \in G'.\text{rb}$.

In case (a) we know there exist $\text{dsb} \in DSB_{\text{t}11}$, $\text{wb} \in WB$ such that $\text{loc}(\text{wb}) = \text{loc}(\text{imp}(a))$, and $\text{imp}(a) \xrightarrow{G.\text{po}} \text{wb} \xrightarrow{G.\text{po}} \text{dsb} \xrightarrow{G.\text{po}} \text{imp}(b)$; thus from the PARMv8-consistency of $G$ we have: $(\text{imp}(a), \text{imp}(b)) \in G.\text{nvo}$. Consequently, from the definition of $G'$ we have $(a, b) \in G'.\text{nvo}$.

In case (b) from Lemma 4 we have $\theta(\xi_a).\text{wu}_x \xrightarrow{G.\text{ob}} \theta(\xi_b).\text{rl}_x$. Moreover, we know there exist $\text{dsb} \in DSB_{\text{t}11}$, $\text{wb} \in WB$ such that $\text{loc}(\text{wb}) = \text{loc}(\text{imp}(a))$, and $\text{imp}(a) \xrightarrow{G.\text{po}} \text{wb} \xrightarrow{G.\text{po}} \text{dsb} \xrightarrow{G.\text{po}} \theta(\xi_a).\text{wu}_x$. As such, from the PARMv8-consistency of $G$ we have: $(\text{imp}(a), \theta(\xi_a).\text{wu}_x) \in G.\text{nvo}$. Moreover, from the PARMv8-consistency of $G$ and since $\theta(\xi_a).\text{wu}_x \xrightarrow{G.\text{ob}} \theta(\xi_b).\text{rl}_x$, we have $\theta(\xi_a).\text{wu}_x \xrightarrow{G.\text{mo}} \theta(\xi_b).\text{rl}_x$ and thus $\theta(\xi_a).\text{wu}_x \xrightarrow{G.\text{mo}} \theta(\xi_b).\text{rl}_x$. As such, we have $(\text{imp}(a), \theta(\xi_a).\text{wu}_x) \in G.\text{nvo}$. Consequently, from the definition of $G'$ we have $(a, b) \in G'.\text{nvo}$.

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

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