Write-observation and Read-preservation TM Correctness Invariants
(Appendix)

Mohsen Lesani         Jens Palsberg
Computer Science Department
University of California, Los Angeles
{lesani, palsberg}@ucla.edu

Contents

1 Proof of Marking Theorem 2
2 TL2 Marking 15
3 DSTM (visible reads) Marking 17
4 Opacity 19
1 Proof of Marking Theorem

For the sake of brevity, we use the shorthand notation
\( \exists l = o . n_T(v_1) ; v_2 \in X \)
for
\( \exists l \in X : obj_X(l) = o \land name_X(l) = n \land thread_X(l) = T \land arg_X(l) = v_1 \land retv_X(l) = v_2 \)
and similarly for universal quantification.

We also use \( W, R \) to denote labels.

Lemma 1. For all \( S \in T_{Sequential}, \; T \in S, \; S' = Visible(S, T), \) and \( T', T'' \in S', \) we have \( T' \preceq_S T'' \iff T' \preceq_S T''. \)

Proof.

\[
T' \preceq_S T''
\iff S'|T' \prec_S S'|T'' \lor T' = T''
\iff S|T' \prec_S S|T'' \lor T' = T''
\iff S|T' \prec_S S|T'' \lor T' = T''
\iff T' \preceq_S T''
\]

In these four steps we apply:
1) the definition of \( \preceq_S \),
2) that the definition of \( Visible(S, T) \) implies both \( S'|T' = S|T' \) and \( S'|T'' = S|T'' \),
3) \( S' \subseteq S \), and
4) the definition of \( \preceq_S \).
Lemma 2. For all $S \in T_{Sequential}$, $T \in S$, $i \in I$, $v, v' \in V$, $R = read_T(i) \in GlobalReads(S)$, $S' = \text{Visible}(S, T)$, $T' \in S'$, and $W' = write_T(i, v') \in GlobalWrites(S)$, we have

$$\text{NoWriteBetween}_{(S'|i)}(W', R) \iff \text{NoWriterBetween}_{S,i}(T', \preceq_S, T)$$

Proof.

$$\text{NoWriteBetween}_{(S'|i)}(W', R) \iff \forall W'' \in Writes(S'|i): W'' \preceq_{(S'|i)} W' \vee R \preceq_{(S'|i)} W''$$

$$\iff \forall T'' \in S'|i: \forall i' \in I: \forall v'' \in V: \forall W'' = write_{T''}(i', v'') \in S'|i: W'' \preceq_{(S'|i)} W' \vee R \preceq_{(S'|i)} W''$$

$$\iff \forall T'' \in S'|i: \forall v'' \in V: \forall W'' = write_{T''}(i, v'') \in S'|i: W'' \preceq_{S'} W' \vee R \preceq_{S'} W''$$

$$\iff \forall T'' \in S': \forall v'' \in V: \forall W'' = write_{T''}(i, v'') \in S': W'' \preceq_S W' \vee R \preceq_S W''$$

$$\iff \forall T'' \in S': \forall v'' \in V: \forall W'' = write_{T''}(i, v'') \in S': W'' \preceq_S W' \vee R \preceq_S W''$$

$$\iff \forall T'' \in S': \forall v'' \in V: \forall W'' = write_{T''}(i, v'') \in S': [T'' = T] \vee (T'' \prec_S T \wedge T'' \in \text{Committed}(S)) \wedge [T'' \preceq_T T] \Rightarrow T'' \preceq_S T'$$

$$\iff \forall T'' \in S': \forall v'' \in V: \forall W'' = write_{T''}(i, v'') \in S': (T'' \in \text{Committed}(S) \wedge T'' \prec_S T) \Rightarrow T'' \preceq_S T'$$

$$\iff \forall T'' \in Writers_S(i): T'' \prec_S T \Rightarrow T'' \preceq_S T'$$

$$\iff \forall T'' \in Writers_S(i): T'' \preceq_S T' \Rightarrow \text{NoWriterBetween}_{S,i}(T', \prec_S, T)$$

In these twelve steps, we apply:

1) the definition of $\text{NoWriteBetween}$,
2) the definition of $\text{Writes}$,
3) the definition of projection $S'|i$,
4) $R$, $W'$ and $W''$ access location $i$,
5) $S' \in T_{Sequential}$ and $R \in GlobalReads(S')$ and $W' \in GlobalWrites(S')$ (that are concluded from $S \in T_{Sequential}$, $R \in GlobalReads(S)$, $W \in GlobalWrites(S)$ and $S' = \text{Visible}(S, T)$),
6) Lemma 1,
7) Boolean logic and that $\preceq_S$ is total,
8) the definition of $\text{Visible}$,
9) logical simplification,
10) the definition of $\text{Writers}$,
11) Boolean logic and that $\preceq_S$ is total, and
12) the definition of $\text{NoWriteBetween}$.

\qed
Lemma 3. $T_{\text{Sequential}} \subseteq \text{Sequential}$

Proof. Straightforward from definitions of $T_{\text{Sequential}}$, $T_{\text{History}}$ and $\text{Sequential}$.

Lemma 4. $\forall i \in I: \forall v, v' \in V: \forall T, T' \in \text{Trans}: \text{if } R = \text{read}_T(i); v, W = \text{write}_T(i, v), W' = \text{write}_{T'}(i, v'), S \in T_{\text{Sequential}}, W \prec_S R, \text{NoWriteBetween}_S(W, R) \text{ and } W' \prec_S R$, then $T = T'$.

Proof. Suppose (1) $S \in T_{\text{Sequential}}$, (2) $W \prec_S R$, (3) $\text{NoWriteBetween}_S(W, R)$ and (4) $W' \prec_S R$. From [1] and Lemma 3, we have (5) $S \in \text{Sequential}$. From [4] and [5], we have (6) $\neg (R \prec_S W')$. From [3] we have (7) $W' \preceq_S W \vee R \prec_S W'$. From [6] and [7], we have (8) $W' \preceq_S W$. From [2] and [8], we have (9) $W' \preceq_S W \preceq_S R$. From [9], [1], and that $W'$ and $R$ are by $T$ and $W$ is by $T'$, we have $T = T'$.
Lemma 5. Suppose $S \in T_{Sequential}$. We have:

$\forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists T' \in Visible(S, T): \exists W = write_T(i, v) \in Visible(S, T):$

$W \prec_{(Visible(S, T) \mid i)} R \land NoWriteBetween_{(Visible(S, T) \mid i)}(W, R)$

$\iff S \in LocalTSeqSpec$

Proof. Suppose $S \in T_{Sequential}$. Thus, from Lemma 3, we have $S \in Sequential$. Let $S' = Visible(S, T)$. From $S \in T_{Sequential}$ and Lemma 1, we have $S' \in T_{Sequential}$. Thus, from Lemma 3, we have $S' \in Sequential$. From the definition of Visible, we have $S'|T = S|T$.

$\forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists T' \in S': \exists W = write_T(i, v) \in S':$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$

$\iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists v' \in V: \exists W' = write_T(i, v') \in S': W' \prec_{S} R \land$

$\exists T' \in S': \exists W = write_T(i, v) \in S':$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$

$\iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists v' \in V: \exists W' = write_T(i, v') \in S': W' \prec_{S} R \land$

$\exists T' \in S': \exists W = write_T(i, v) \in S':$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$

$\iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists v' \in V: \exists W' = write_T(i, v') \in S': W' \prec_{(S' \mid i)} R \land$

$\exists T' \in S': \exists W = write_T(i, v) \in S':$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$

$\iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists W = write_T(i, v) \in S'$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$

$\iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S):$

$\exists W = write_T(i, v) \in S'$

$W \prec_{(S' \mid i)} R \land NoWriteBetween_{(S' \mid i)}(W, R)$
\[ \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S'} R \land \text{NoWriteBetween}_{(S' | i)}(W, R) \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \text{NoWriteBetween}_{(S' | i)}(W, R) \]
\[ \iff \forall T \in S: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S' | i): W' \preceq_{(S' | i)} W \lor R \prec_{(S' | i)} W' \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S' | i): W' \preceq_{(S' | i)} W \lor -(R \prec_{(S' | i)} W') \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S' | i): W \prec_{(S' | i)} W' \prec_{(S' | i)} R \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v') \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S | i): W \prec_{(S | i)} W' \prec_{(S | i)} R \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S | i): -(W \prec_{(S | i)} W') \lor -(R \prec_{(S | i)} W') \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \forall W' \in \text{Writes}(S | i): W' \preceq_{(S | i)} W \lor R \prec_{(S | i)} W' \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S'} R \land \text{NoWriteBetween}_{(S' | i)}(W, R) \]
\[ \iff \forall T \in S: \forall i \in I: \forall v \in V: \forall R = read_T(i): v \in LocalReads(S): \]
\[ \exists W = write_T(i, v) \in S: \]
\[ W \prec_{S} R \land \text{NoWriteBetween}_{(S | i)}(W, R) \]
\[ \iff S \in \text{LocalTSeqSpec} \]

In these twenty steps, we apply: 1) the definition of LocalReads,
2) the definition of Visible,
3) $S'|T = S|T$ and that both $W'$ and $R$ are by $T$,
4) that both $W'$ and $R$ are on $i$,
5) Lemma 4,
6) duplicate conjunction,
7) the definition of Visible,
8) that both $R$ and $W$ are on $i$,
9) $S'|T = S|T$ and that both $R$ and $W$ are by $T$,
10) the definition of NoWriteBetween,
11) first-order logic,
12) $(S' | i) \in \text{Sequential},$
13) from $(S' | i) \in T\text{Sequential}$, $R$ and $W$ are by transaction $T$ and $W'$ is between them, we have $W'$ is by $T$,
14) $S'|T = S|T$,
15) from $(S | i) \in T\text{Sequential}$, $R$ and $W$ are by transaction $T$ and $W'$ is between them, we have $W'$ is by $T$.
16) first-order logic,
17) $(S | i) \in \text{Sequential},$
18) $(S | i) \in \text{Sequential}$, $\text{thread}_H(R) = \text{thread}_H(W) = T$ and $\text{arg}_1(H)(R) = \text{arg}_1(H)(W) = i$,
19) the definition of NoWriteBetween,
20) the definition of $\text{LocalTSeqSpec}$. 

\Box
Lemma 6. Suppose $S \in T_{\text{Sequential}} \cap T_{\text{Complete}}$. We have:

$$S \in T_{\text{SeqSpec}}$$

$$\iff \forall T \in S: \forall i \in I: (\text{Visible}(S, T) \mid i) \in \text{SeqSpec}(i)$$

$$\iff \forall T \in S: \forall i \in I:$$

$$\forall T'' \in (\text{Visible}(S, T) \mid i): \forall v \in V: \forall R = \text{read}_{T''}(i): v \in (\text{Visible}(S, T) \mid i):$$

$$\exists T' \in \text{Visible}(S, T): \exists W = \text{write}_{T'}(i, v) \in \text{Visible}(S, T):$$

$$W \prec (\text{Visible}(S, T) \mid i) R \land \text{NoWriteBetween}_{\text{Visible}(S, T) \mid i}(W, R)$$

$$\iff \forall T \in S: \forall i \in I:$$

$$\forall T'' \in \text{Visible}(S, T): \forall v \in V: \forall R = \text{read}_{T''}(i): v \in \text{Visible}(S, T):$$

$$\exists T' \in \text{Visible}(S, T): \exists W = \text{write}_{T'}(i, v) \in \text{Visible}(S, T):$$

$$W \prec (\text{Visible}(S, T) \mid i) R \land \text{NoWriteBetween}_{\text{Visible}(S, T) \mid i}(W, R)$$

$$\iff \forall T \in S: \forall i \in I:$$

$$\forall T'' \in \text{Visible}(S, T): \forall v \in V: \forall R = \text{read}_{T''}(i): v \in \text{LocalReads}(S):$$

$$\exists T' \in \text{Visible}(S, T): \exists W = \text{write}_{T'}(i, v) \in \text{Visible}(S, T):$$

$$W \prec (\text{Visible}(S, T) \mid i) R \land \text{NoWriteBetween}_{\text{Visible}(S, T) \mid i}(W, R)$$

$$\land$$

$$\forall T \in S: \forall i \in I:$$

$$\forall R = \text{read}_{T}(i): v \in \text{GlobalReads}(S):$$

$$\exists T' \in \text{Visible}(S, T): \exists W = \text{write}_{T'}(i, v) \in \text{Visible}(S, T):$$

$$W \prec (\text{Visible}(S, T) \mid i) R \land \text{NoWriteBetween}_{\text{Visible}(S, T) \mid i}(W, R)$$

$$\iff S \in \text{LocalTSeqSpec} \land$$

$$\forall T \in S: \forall i \in I:$$

$$\forall R = \text{read}_{T}(i): v \in \text{GlobalReads}(S):$$

$$\exists T' \in \text{Visible}(S, T): \exists W = \text{write}_{T'}(i, v) \in \text{Visible}(S, T):$$

$$W \prec (\text{Visible}(S, T) \mid i) R \land \text{NoWriteBetween}_{\text{Visible}(S, T) \mid i}(W, R)$$
In these thirteen steps, we apply:
1) the definition of $T_{SeqSpec}$ and $S \in T_{Sequential} \cap T_{Complete}$,
2) the definition of $SeqSpec(i)$,
3) $R$ and $W$ access location $i$,
4) that we can choose $T'' = T$,
5) $Reads(S) = LocalReads(S) \cup GlobalReads(S)$,
6) Lemma 5,
7) that $R$ and $W$ are both on location $i$,
8) that $R$ and $W$ are by transactions $T$ and $T'$ respectively, $Visible(S,T) \in T_{Sequential}$, and $R \in GlobalReads(Visible(S,T))$ (because $R \in GlobalReads(R)$ and $Visible(S,T)|T = S|T$),
9) Lemma 1,
10) $T' \ll S T$ and $NoWriteBetween(Visible(S,T) | i)(W, R)$,
11) Lemma 2,
12) $T' \in \text{Visible}(S,T)$ and $(T' \prec_S T)$, and
13) the definition of $\text{Visible}(S,T)$. 

\square
Lemma 7. (Invariance) If $H \equiv H'$, then $\text{Marking}(H) = \text{Marking}(H')$ and $\text{ReadPres}(H) = \text{ReadPres}(H')$ and $\text{WriteObs}(H) = \text{WriteObs}(H')$.

Proof. Immediate from the definitions of $\text{Marking}$, $\text{ReadPres}$, and $\text{WriteObs}$. \hfill \Box

Lemma 8. \( \forall H \in T\text{History} : \forall \sqsubseteq \in \text{Marking}(H) : \exists S \in T\text{Sequential} : H \equiv S \land \preceq_H \subseteq \preceq_S \land \preceq_S \subseteq \sqsubseteq. \)

Proof. Let $H \in T\text{History}$ and let $\sqsubseteq \in \text{Marking}(H)$. We have that $\sqsubseteq$ is a total order of $\text{Trans}$ so we can choose a permutation $\pi$ on $1..n$ such that $\forall i, j \in 1..n : (i < j) \iff (T_{\pi(i)} \sqsubseteq T_{\pi(j)})$. Define: $S = H[T_{\pi(1)}, \ldots, H[T_{\pi(n)}]$. It is straightforward to prove that $S \in T\text{Sequential} \land H \equiv S \land \preceq_H \subseteq \preceq_S \land \preceq_S \subseteq \sqsubseteq. \hfill \Box

Lemma 9. Suppose $\sqsubseteq \in \text{Marking}(H) \land p_2 \notin \text{Writers}_H(i)$.
If $\text{NoWriterBetween}_{H,i}(T_1, \sqsubseteq, p_2)$ and $\text{NoWriterBetween}_{H,i}(p_2, \sqsubseteq, T_3)$,
them$\text{NoWriterBetween}_{H,i}(T_1, \sqsubseteq, T_3)$.

Proof.

\[
\text{NoWriterBetween}_{H,i}(T_1, \sqsubseteq, p_2) \land \text{NoWriterBetween}_{H,i}(p_2, \sqsubseteq, T_3) \\
\iff \forall T \in \text{Writers}_H(i) : (T \sqsubseteq T_1 \lor p_2 \sqsubseteq T) \land (T \sqsubseteq p_2 \lor T_3 \sqsubseteq T) \\
\iff \forall T \in \text{Writers}_H(i) : (T \sqsubseteq T_1 \land (T \sqsubseteq p_2 \lor T_3 \sqsubseteq T)) \lor \\
\quad (p_2 \sqsubseteq T \land T \sqsubseteq p_2) \lor (p_2 \sqsubseteq T \land T_3 \sqsubseteq T) \\
\iff \forall T \in \text{Writers}_H(i) : (T \sqsubseteq T_1) \lor (T_3 \sqsubseteq T) \\
\iff \text{NoWriterBetween}_{H,i}(T_1, \sqsubseteq, T_3)
\]

The first step uses the definition of $\text{NoWriterBetween}$. The second step uses $\land$ distribution over $\lor$. The third step simplifies the first disjunct using conjunction elimination, eliminates the second disjunct using $p_2 \notin \text{Writers}_H(i)$ and simplifies the third disjunct using conjunction elimination. The fourth step uses the definition of $\text{NoWriterBetween}$. \hfill \Box
Lemma 10. Suppose \( S \in T_{\text{Sequential}} \cap T_{\text{Complete}} \). We have:

\[
S \in T_{\text{SeqSpec}} \iff S \in \text{Markable}
\]

Proof. Let \( S \in T_{\text{Sequential}} \cap T_{\text{Complete}} \). From Lemma 6, the definition of Markable, and \( S \in T_{\text{Complete}} \), we have that we must prove:

\[
S \in \text{LocalTSeqSpec} \land \\
\forall T \in S: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i):v \in \text{GlobalReads}(S): \\
\exists T' \in \text{Committed}(S): \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
(T' \prec_S T) \land \text{NoWriterBetween}_{S,i}(T', \preceq_S T) \\
\iff \exists \subseteq \in \text{Marking}(S): \not\subseteq S \subseteq \subseteq \land \subseteq \in \text{ReadPres}(S) \land \subseteq \in \text{WriteObs}(S)
\]

From the definition of WriteObs and LastPreAccessor we have that:

\[
\subseteq \in \text{WriteObs}(S) \\
\iff S \in \text{LocalTSeqSpec} \land \\
\forall T \in \text{Trans}: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i):v \in \text{GlobalReads}(S): \\
\exists T' \in \text{Trans}: \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
T' \in \text{Writers}_{S,i}(T') \land T' \neq T \land T' \sqsubset R \land \text{NoWriterBetween}_{S,i}(T', \sqsubseteq, R)
\]

We are now ready to prove the two directions of the equivalence.

\( \Rightarrow: \)

Assume that

\[
S \in \text{LocalTSeqSpec} \land \\
\forall T \in S: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i):v \in \text{GlobalReads}(S): \\
\exists T' \in \text{Committed}(S): \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
(T' \prec_S T) \land \text{NoWriterBetween}_{S,i}(T', \preceq_S T)
\]

Define:

\[
p_1 \sqsubseteq p_2 \iff (p_1 \prec_S p_2) \lor \\
(\text{thread}_{S}(p_1) \preceq_S p_2) \lor \\
(p_1 \preceq_S \text{thread}_{S}(p_2))
\]

\[
p_1 \sqsubseteq p_2 \iff p_1 \sqsubseteq \lor p_2 p_1 = p_2
\]

We show that

\[
\subseteq \in \text{Marking}(S) \land \\
\not\subseteq S \subseteq \subseteq \land \subseteq \in \text{ReadPres}(S) \land \\
S \in \text{LocalTSeqSpec} \land \\
\forall T \in \text{Trans}: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i):v \in \text{GlobalReads}(S): \\
\exists T' \in \text{Trans}: \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
T' \in \text{Committed}(S) \land T' \neq T \land T' \sqsubset R \land \text{NoWriterBetween}_{S,i}(T', \sqsubseteq, R)
\]
It is straightforward to prove $\square \in \mathit{Marking}(S)$ and $\preceq_S \subseteq \square$, $\square \in \mathit{ReadPres}(S)$. Additionally, the first conjunct of $\mathit{WriteObs}(S)$ (that is, $S \in \mathit{LocalTSeqSpec}$) is immediate from the assumption. So, we still need to prove the second conjunct of $\mathit{WriteObs}(S)$.

Let $T \in \text{Trans}$, $i \in I$, $v \in V$, $R = \text{read}_T(i)\cdot v \in \text{GlobalReads}(S)$. From the assumption (the left-hand side), we have that we can find (1) $T' \in \mathit{Committed}(S)$ and (2) $W = \text{write}_T(i, v) \in \text{GlobalWrites}(S)$ such that (3) $(T' \not\preceq_S T)$ and (4) $\mathit{NoWriterBetween}_{S,i}(T', \preceq_S, T)$. Let us now prove each conjunct of $T' \neq T \land T' \subseteq R \land \mathit{NoWriterBetween}_{S,i}(T', \subsetneq, R)$ in turn.

From [3] and that $\preceq_S$ is a total order of $\text{Trans}(S)$, we have (5) $T' \neq T$. From [3] and the definition of $\subseteq$, we have $T' \subseteq R$. From [4] and $\preceq_S \subseteq \subseteq$, we have (6) $\mathit{NoWriterBetween}_{S,i}(T', \subsetneq, \subseteq, R)$. From $T \preceq_S T$ and the definition of $\subseteq$, we have (7) $R \subseteq T$. From [6], [7] and the definition of $\subseteq$ and transitivity of $\preceq_S$, we have $\mathit{NoWriterBetween}_{S,i}(T', \subsetneq, R)$.

\[\Leftarrow: \]

Assume the right-hand side and choose $\square \in \mathit{Marking}(S)$ such that:

\[
\preceq_S \subseteq \subseteq \land \square \in \mathit{ReadPres}(S) \land \\
S \in T\mathit{LocalSeqSpec} \land \\
\forall T \in \text{Trans}: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i)\cdot v \in \text{GlobalReads}(S): \\
\exists T' \in \mathit{Committed}(S): \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
T' \neq T \land T' \subseteq R \land \mathit{NoWriterBetween}_{S,i}(T', \subsetneq, R)
\]

We show that

\[
S \in \mathit{LocalTSeqSpec} \land \\
\forall T \in S: \forall i \in I: \forall v \in V: \forall R = \text{read}_T(i)\cdot v \in \text{GlobalReads}(S): \\
\exists T' \in \mathit{Committed}(S): \exists W = \text{write}_T(i, v) \in \text{GlobalWrites}(S): \\
(T' \not\preceq_S T) \land \mathit{NoWriterBetween}_{S,i}(T', \preceq_S, T)
\]

The first conjunct (of the left-hand side), $S \in \mathit{LocalTSeqSpec}$, is immediate from the assumption. From the assumption we have (1) $\preceq_S \subseteq \subseteq$, (2) $\subseteq \in \mathit{ReadPres}(S)$. Let $T \in \text{Trans}$, $i \in I$, $v \in V$, $R = \text{read}_T(i)\cdot v \in \text{GlobalReads}(S)$. From the above property of $\subseteq$, we have that we can find (3) $T' \in \mathit{Committed}(S)$ and (4) $W = \text{write}_T(i, v) \in \text{GlobalWrites}(S)$ such that (5) $T' \neq T$ and (6) $T' \subseteq R$ and (7) $\mathit{NoWriterBetween}_{S,i}(T', \subsetneq, \subseteq, R)$. From [1], that $\subseteq$ is a total order on $\text{Trans}(S)$ ($\subseteq \in \mathit{Marking}(S)$), and that $\preceq_S$ is a total order on $\mathit{Trans}(S)$ ($S \in T\mathit{Sequential}$), we have (8) $\forall T, T' \in \text{Trans}: T' \subseteq T \Rightarrow T' \not\preceq_S T$.

First we prove $T' \not\preceq_S T$. From [2], we have (9) $\mathit{NoWriterBetween}_{S,i}(T', \subsetneq, \subseteq, R)$. From [3] and [4], we have (10) $T' \in \mathit{Writers}_S(i)$. From [9] and [10], we have (11) $T' \subseteq T \lor R \subseteq T'$. From [6], $T' \neq R$ and $\subseteq$ is a total order on $\{R\} \cup \mathit{Writers}_S(i)$ ($\subseteq \in \mathit{Marking}(S)$), we have (12) $R \not\subseteq T'$. From [11] and [12], we have (13) $T' \subseteq T$. From [8] and [13], we have (14) $T' \not\preceq_S T$. From [14] and [5], we have $T' \not\preceq_S T$.

Second, we prove $\mathit{NoWriterBetween}_{S,i}(T', \preceq_S, T)$. From [2], we have (15) $\mathit{NoWriterBetween}_{S,i}(R, \subsetneq, T)$. From $R \not\in \mathit{Writers}_S(i)$, [7], [15], and Lemma 9, we have (16) $\mathit{NoWriterBetween}_{S,i}(T', \subsetneq, T)$. From [16] and [8] we have $\mathit{NoWriterBetween}_{S,i}(T', \preceq_S, T)$. \\[\square\]
Theorem (Marking)  \( \text{FinalStateOpaque} = \text{Markable} \).

Proof.

\[
\text{FinalStateOpaque} = \{ H \in \text{THistory} \mid \exists H' \in \text{TExtension}(H) : \exists S \in \text{TSequential} : \\
H' \equiv S \land \mathcal{S}_{H'} \subseteq \mathcal{S} \land S \in \text{TSeqSpec} \}
\]

\[
= \{ H \in \text{THistory} \mid \exists H' \in \text{TExtension}(H) : \exists S \in \text{TSequential} : \\
H' \equiv S \land \mathcal{S}_{H'} \subseteq \mathcal{S} \land S \in \text{Markable} \}
\]

In these eight steps we apply:
1) the definition of \( \text{FinalStateOpaque} \),
2) Lemma 10 and \( S \in \text{TComplete} \) (because \( H' \in \text{TExtension}(H) \) and \( H' \equiv S \)),
3) the definition of \( \text{Markable} \) and \( S \in \text{TComplete} \),
4) Lemma 7,
5) logical rearrangement,
6) transitivity of \( \subseteq \),
7) Lemma 8, and
8) the definition of \( \text{Markable} \). \( \square \)
## TL2 Marking

### Shared objects:
- \( r: \text{SafeReg}[I], \text{initially } \bot \)
- \( \text{ver}: \text{AtomicReg}[I], \text{initially } 0 \)
- \( \text{lock}: \text{TryLock}[I], \text{initially } \mathbb{R} \)
- \( \text{clock}: \text{SCounter}, \text{initially } 0 \)

### Thread-local objects:
For each \( T \in \text{Trans} \):
- \( \text{rver}_T: \text{SafeReg}, \text{initially } \bot \)
- \( \text{rset}_T: \text{BasicSet}, \text{initially } \emptyset \)
- \( \text{wset}_T: \text{BasicMap}, \text{initially } \emptyset \)

<table>
<thead>
<tr>
<th>R01</th>
<th>def read(_T)(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R02</td>
<td>if (rver(_T = \bot))</td>
</tr>
<tr>
<td>R03</td>
<td>snap := clock.read()</td>
</tr>
<tr>
<td>R04</td>
<td>rver(_T.write(snap))</td>
</tr>
</tbody>
</table>

| R05 | if (i \(\in\) dom(wset\(_T\))) |
| R06 | return wset\(_T\)(i) |

| R07 | t := ver\(_[i].read()\) |
| R08 | v := reg\(_[i].read()\) |
| R09 | l := lock\([i].read()\) |
| R10 | t' := ver\(_[i].read()\) |
| R11 | if(\(\neg(l = false \land t = t' \land t' \leq rver_T)\)) |
| R12 | rver\(_T\.add(i)\) |
| R13 | return A |

| W01 | def write\(_T\)(i,v) |
| W02 | wset\(_T\.put(i \mapsto v)\) |
| W03 | return ok |

| C01 | def commit\(_T\) |
| C02 | foreach (i \(\in\) dom(wset\(_T\))) |
| C03 | locked := lock\([i].trylock() |
| C04 | if (locked) |
| C05 | lset.add(i) |
| C06 | else |
| C07 | foreach (i \(\in\) lset) lock\([i].unlock() |
| C08 | return A |

| C09 | wver := clock.iaf |
| C10 | if (wver \(!=\) rver\(_T + 1\)) |
| C11 | foreach (i \(\in\) rset\(_T\)) |
| C12 | l := lock\([i].read() |
| C13 | t := ver\(_[i].read() |
| C14 | if (\(\neg(l = false \land t \leq rver_T)\)) |
| C15 | foreach (i \(\in\) lset) lock\([i].unlock() |
| C16 | return A |

| C17 | foreach ((i \mapsto v) \(\in\) wset\(_T\) |
| C18 | reg\([i].write(v) |
| C19 | ver\([i].write(wver) |
| C20 | lock\([i].unlock() |
| C21 | return C |

In addition to the orders imposed by the data and control dependencies and lock synchronization, the following orders are required: \( R06 \prec R07, R07 \prec R08, R08 \prec R09, C12 \prec C13, C18 \prec C19 \)

Figure 1: TL2 Algorithm
Consider an execution history $X$ of TL2 such that $H = X|mem$ and $H \in TComplete$. Let

$$\begin{align*}
\text{readAcc}(R) &= R08 \text{ in } R \\
\text{writeAcc}(T, i) &= C18 \text{ for } i \text{ in } \text{Commit}_T \\
\text{Eff}(T) &= \begin{cases} R03 \text{ (in the first read of } T) & \text{ if } T \in \text{Aborted}(H) \\
C09 \text{ (in } \text{commit}_T) & \text{ if } T \in \text{Committed}(H) \end{cases}
\end{align*}$$

Let $\prec_{\text{clock}}$ represent the linearization order of the strong counter $\text{clock}$. The marking $\sqsubseteq$ for $H$ is the reflexive closure of $\sqsubset$ that is defined as follows:

Let $T, T' \in \text{Trans}(H)$:

$$T \sqsubseteq T' \iff \text{Eff}(T) \prec_{\text{clock}} \text{Eff}(T')$$

Let $R \in \text{Reads}(H), i = \text{arg1}(R), T \in \text{Writers}_H(i)$:

$$T \sqsubseteq R \iff \text{writeAcc}(T, i) \not\preceq_X \text{readAcc}(R)$$

$$R \sqsubseteq T \iff \text{readAcc}(R) \not\preceq_X \text{writeAcc}(T, i)$$

Figure 2: The marking of TL2.

The marking relation for TL2 is defined in Figure 2. The effect order of transactions is the linearization order of their calls to the $\text{clock}$ strong counter. The access order of read operations and writer transactions to location $i$ is the execution order of their access to the $\text{reg}[i]$ register.
3 DSTM (visible reads) Marking

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shared objects:</td>
</tr>
<tr>
<td></td>
<td>state: CASReg[Trans], initially R</td>
</tr>
<tr>
<td></td>
<td>ref: CASReg[i], initially new Loc(T0, Ø, 0, 0)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>R01</td>
<td>def readT(i)</td>
</tr>
<tr>
<td>R02</td>
<td>r := ref[i].read()</td>
</tr>
<tr>
<td>R03</td>
<td>v := currentValueT(r)</td>
</tr>
<tr>
<td>R04</td>
<td>r' = r.clone()</td>
</tr>
<tr>
<td>R05</td>
<td>r'.rset.add(T)</td>
</tr>
<tr>
<td>R06</td>
<td>b := ref[i].cas(r, r')</td>
</tr>
<tr>
<td>R07</td>
<td>s := stateT.read()</td>
</tr>
<tr>
<td>R08</td>
<td>if (¬b ∨ (s = A))</td>
</tr>
<tr>
<td>R09</td>
<td>return A</td>
</tr>
<tr>
<td>R10</td>
<td>else</td>
</tr>
<tr>
<td>R11</td>
<td>return v</td>
</tr>
</tbody>
</table>

| C01 | def commitT()                                                    |
| C02 | b := stateT.cas(R, C)                                            |
| C03 | if (b)                                                           |
| C04 | return C                                                         |
| C05 | else                                                             |
| C06 | return A                                                         |

| W01 | def writeT(i, v)                                                 |
| W02 | r := ref[i].read()                                               |
| W03 | w := r.writer.read()                                            |
| W04 | if (w = T)                                                       |
| W05 | r.newVal.write(v)                                               |
| W06 | return ok                                                       |
| W07 | v' := currentValueT(r)                                           |
| W08 | foreach (T' ∈ r.rset)                                           |
| W09 | stateT'.cas(R, A)                                                |
| W10 | r' := new Loc(T, Ø, v', v)                                      |
| W11 | b := ref[i].cas(r, r')                                           |
| W12 | if (b)                                                           |
| W13 | return ok                                                       |
| W14 | else                                                            |
| W15 | return A                                                        |

| V01 | def currentValueT(r)                                            |
| V02 | T' = r.writer.read()                                            |
| V03 | if (¬(T' = T))                                                  |
| V04 | stateT'.cas(R, A)                                               |
| V05 | s := stateT.read()                                              |
| V06 | if (s = A)                                                       |
| V07 | return r.oldVal                                                 |
| V08 | else                                                            |
| V10 | return r.newVal                                                 |

Figure 3: DSTM (visible reads) Algorithm
Consider an execution history $X$ of DSTM such that $H = X|\text{mem}$ and $H \in TComplete$. Let

\[
\begin{align*}
\text{readAcc}(R) &= R06 \text{ in } R \\
\text{writeAcc}(T,i) &= W12 \text{ in the first write to } i \text{ by } T \\
\text{Eff}(T) &= \begin{cases} 
C02 \text{ of the commit operation} & \text{if } T \text{ is committed} \\
R06 \text{ of the last successful read} & \text{if } T \text{ is aborted and has a successful read} \\
\text{Any point in } T & \text{if } T \text{ is aborted and has no successful read}
\end{cases}
\end{align*}
\]

Let $\preceq_{\text{ref}[i]}$ represent the linearization order of $\text{ref}[i]$. The marking $\sqsubseteq$ for $H$ is the reflexive closure of $\sqsubseteq$ that is define as follows:

Let $T, T' \in \text{Trans}(H)$:

$\quad T \sqsubseteq T' \iff \text{Eff}(T) \preceq_{X} \text{Eff}(T')$

Let $R \in \text{Reads}(H), i = \text{arg1}(R), T \in \text{Writers}_{H}(i)$:

$\quad T \sqsubseteq R \iff \text{writeAcc}(T,i) \preceq_{\text{ref}[i]} \text{readAcc}(R)$

$\quad R \sqsubseteq T \iff \text{readAcc}(R) \preceq_{\text{ref}[i]} \text{writeAcc}(T,i)$

Figure 4: The marking of DSTM (visible reads).

The marking relation for DSTM (visible reads) is defined in Figure 4.

Committed transactions take effect at the final $\text{cas}$ of their state from $R$ to $C$, $C02$, of their commit operation. Aborted transactions that have successful read operations take effect at state check, $R06$, of their last successful read.

The access order of read operations and writer transactions to location $i$ is the linearization order of their $\text{cas}$ calls to the $\text{ref}[i]$ register.
4 Opacity

\[
\begin{align*}
\text{Reads}(H) &= \{ R | R \in H \land \text{obj}_H(R) = \text{this} \land \\
&\quad \text{name}_H(R) = \text{read} \land \text{retv}_H(R) \neq \emptyset \} \\
\text{Writes}(H) &= \{ W | W \in H \land \text{obj}_H(W) = \text{this} \land \\
&\quad \text{name}_H(W) = \text{write} \land \text{retv}_H(W) \neq \emptyset \} \\
\text{Trans}(H) &= \{ T | \exists l \in H : \text{thread}_H(l) = T \} \\
\text{TSequential} &= \{ S \in \text{THistory} | \preceq_S \text{ is a total order of Trans}(S) \} \\
\text{Committed}(H) &= \{ T | \exists l \in H : \text{thread}_H(l) = T \land \text{retv}_H(l) = \emptyset \} \\
\text{Aborted}(H) &= \{ T | \exists l \in H : \text{thread}_H(l) = T \lor \text{retv}_H(l) = \emptyset \} \\
\text{Completed}(H) &= \text{Committed}(H) \cup \text{Aborted}(H) \\
\text{Live}(H) &= \text{Trans}(H) \setminus \text{Completed}(H) \\
\text{TComplete} &= \{ H \in \text{THistory} | \forall T \in \text{Trans}(H) : T \in \text{Completed}(H) \} \\
\text{CommitPending}(H) &= \{ T \in \text{Live}(H) | \exists l \in H : \text{thread}_H(l) = T \land \text{name}_H(l) = \text{commit} \\
&\quad i\text{Ev}(l) \in H \land \neg (r\text{Ev}(l) \in H) \} \\
\text{TExtension}(H) &= \{ H' \in \text{THistory} | H \text{ is a prefix of } H' \land \forall T \in \text{Trans}(H') \Rightarrow T \in \text{Trans}(H) \land \\
&\quad \text{Live}(H) \setminus \text{CommitPending}(H) \subseteq \text{Aborted}(H') \land \\
&\quad \text{CommitPending}(H) \subseteq \text{Completed}(H') \} \\
\text{Visible}(S,T) &= \text{filter} \{ S, \lambda T'.(T' = T) \lor ((T' \prec_S T) \land T' \in \text{Committed}(S)) \} \\
\text{NoWriteBetween}_S(W,R) &= \forall W' \in \text{Writes}(S) : W' \preceq_S W \lor R \prec_S W' \\
\text{SeqSpec}(i) &= \{ S \in \text{Sequential} | \forall R \in \text{Reads}(S) : \exists W \in \text{Writes}(S) : \\
&\quad W \prec_S R \land \text{NoWriteBetween}_S(W,R) \land \\
&\quad \text{retv}_S(R) = \text{arg}_2 S(W) \} \\
\text{TSeqSpec} &= \{ S \in \text{TSequential} \cap \text{TComplete} | \forall T \in S : \forall i \in I : \\
&\quad \text{Visible}(S,T) | i \in \text{SeqSpec}(i) \} \\
\text{FinalStateOpaque} &= \{ H \in \text{THistory} | \exists H' \in \text{TExtension}(H) : \exists S \in \text{TSequential} : \\
&\quad H' \equiv S \land \preceq_{H'} \subseteq \preceq_S \land S \in \text{TSeqSpec} \}
\end{align*}
\]

Figure 5: FinalStateOpaque

Opacity of a TM algorithm is defined in two steps. First, it is defined what it means for a transaction history to be opaque which is called final-state-opacity. Then, a TM algorithm is defined to be opaque if every transaction history of every source program running on top of that TM algorithm is final-state-opaque.

FinalStateOpaque is defined in Figure 5. We use $T$ prefix before some of the terms to avoid confusion with the terms that we defined above for execution histories of objects. We say that a transaction history is sequential if it is a sequence of transactions. A transaction $T$ is committed or aborted in a transaction history $H$ if there is respectively a commit or abort response event for $T$ in $H$. A completed transaction is either committed or aborted. A live transaction is a transaction that is not completed. A transaction history is complete if all its transactions are completed. A pending transaction has a pending event and a commit-pending transaction has a commit pending event. An extension of a history is obtained by committing or
aborting its commit-pending transactions and aborting the other live transactions. If \( H \) is a transaction history and \( p \) is a predicate on transaction identifiers, we define \( \text{filter}(H, p) \) to be the subsequence of \( H \) that contains the events of transactions \( T \) for which \( p(T) \) is true. The visible history for a transaction \( T \) in a sequential transaction history \( S \), \( \text{Visible}(S, T) \), is the sequence of committed transactions before \( T \) in \( S \) and \( T \) itself. The sequential specification of a location \( i \), \( \text{SeqSpec}(i) \), is the set of sequential histories of read and write method calls on location \( i \) where every read returns the value given as the argument to the latest preceding write (regardless of thread identifiers). It is essentially the sequential specification of a register. Transactional sequential specification is the set of complete sequential transaction histories \( S \) that for every transaction \( T \) and location \( i \), \( \text{Visible}(S, T)|i \) is a member of the sequential specification of \( i \). A transaction history \( H \) is final-state-opaque if there is an equivalent sequential transaction history \( S \) for an extension of \( H \) such that \( S \) is real-time-preserving and a member of transactional sequential specification. The sequential history \( S \) is called the justifying history. In other words, every correct concurrent execution is indistinguishable from a correct sequential execution.